



Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.

Foxborough, MA 02035

Tel: 508-543-8600

FAX: 508-543-1503

E-MAIL: support@iconics.com

WEB: <http://www.iconics.com>

Contents

Getting Started

AlarmWorX32 Container Application

AlarmWorX32 Container: Other Screen Features

Getting Started

About AlarmWorX32

Installing AlarmWorX32 and all of its Components

Starting AlarmWorX32 Container

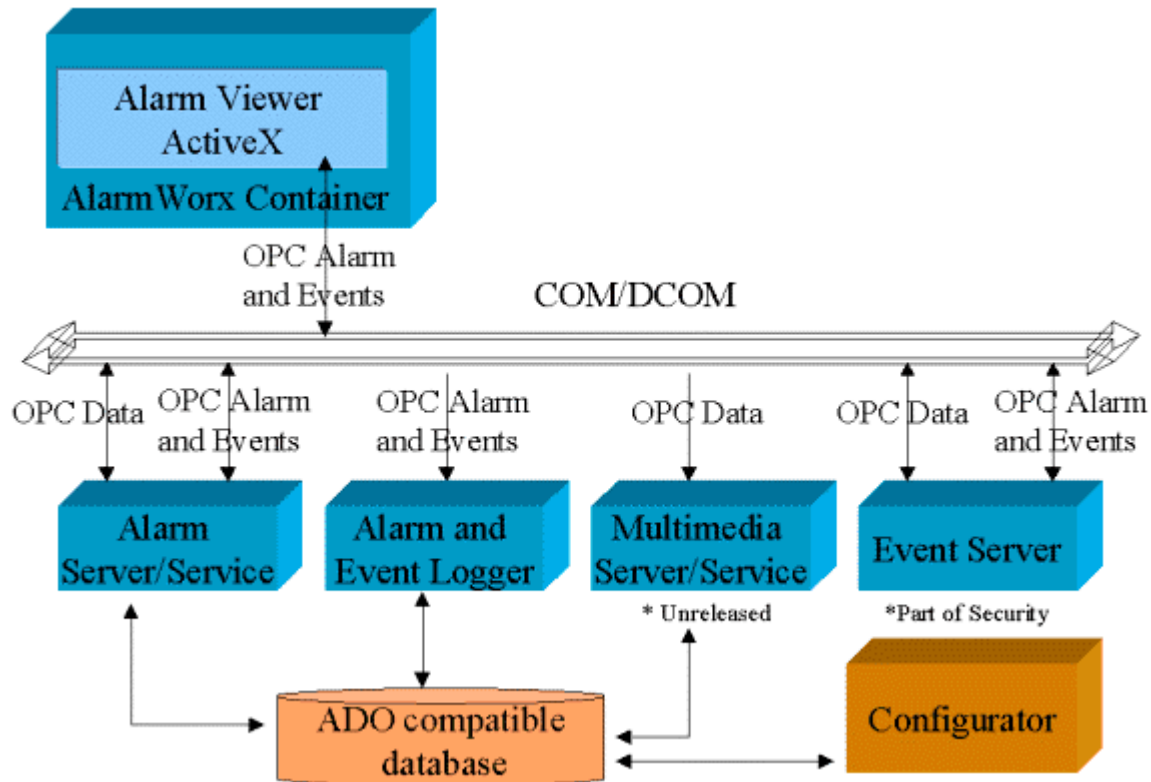
Display Tricks

Subscription/Filtering

Additional Features

About AlarmWorX32

Iconics AlarmWorX32 is a family of modular alarming products. The specific modules dealt with are the Alarm Container, the Alarm Server, the Alarm Logger, the Alarm Viewer, and the Alarm Report. The Alarm Container, titled AlarmWorx32, is an ActiveX container capable of holding various ActiveX components.



AlarmWorX32 Architecture

Alarm Server

The AlarmWorX32 Server receives field data from any OPC Compliant Data Access Server and performs alarm detection and reporting based on the OPC sent to any OPC Alarm & Event clients that subscribe. The AlarmWorX32 Viewer and The AlarmWorX32 Logger are two examples of clients that can receive these notifications from the server.

Alarm Logger

This component logs events and alarms to a database and/or printer as a background process. The logger can run as an executable or service and is able to process alarm information from several alarm servers locally or on a

network. The configuration of the database is user customizable, i.e. it allows autoflushing at a specified period, autostart, circular and append logging modes, etc. Of primary importance to the database issue is performance. The logger is able to handle bursts of alarms due to the design and implementation of the logger's database handling.

Alarm Viewer

The Alarm Viewer is a current event alarm ActiveX. Because this component is an ActiveX, it can be placed in any ActiveX container application, including GraphWorX32, VB or a web page to name a few. The alarm viewer displays current alarm information and handles the user interface to the alarm system (such as alarm acknowledgement). The layout of information displayed, including sort order, color, font, and displayed data, is user configurable.

Alarm Report

The Alarm Report allows reporting (user configured or pre-configured) and graphing of alarms. The source of the Alarm data can be live alarms, alarms previously logged by the Alarm Logger, or a combination of both. The Alarm Report is an ActiveX, allowing it to be placed in any ActiveX container application, including GraphWorX32, VB or a web page to name a few.

Multimedia Server (currently unreleased)

This ActiveX contains the functionality to do multimedia functions (play a video clip, a .wav file, etc.) and TAPI functions such as support for pagers, FAX machines, and the telephony API (TAPI)) when an alarm is received. The ActiveX allows the user to select a particular alarm (or set of alarms) and then the multimedia file to play when that alarm is received or the telephone information needed to dial out when that alarm is received. It also allows configuration of scheduling data for the paging/faxing/calling functions so that this information is used when an alarm comes in to determine whether to page/fax/call and who the receiver would be at any given time.

Getting Started will take you through the Installation and setup of AlarmWorX32. We will discuss some basic steps for starting the AlarmWorX32 Container. For more detailed information on starting, configuring, and running the Container component and each of the other AlarmWorX components, please see corresponding section.

Installing AlarmWorX32 and all of its Components

There are two ways to install AlarmWorX32 and all of its components:

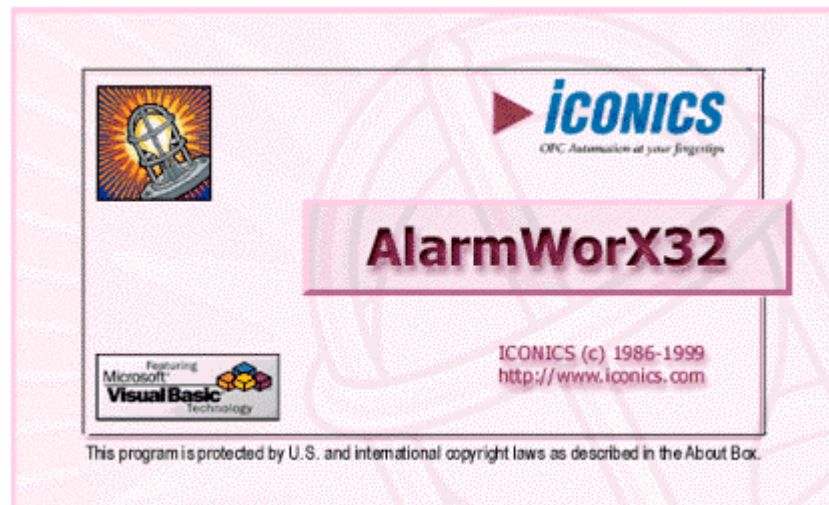
A. Install from the Genesis32 Product CD-ROM.

- Insert CD-ROM into CD drive. The CD should automatically start.
- From the main menu of the CD-ROM select **Software** which will bring up a menu with all of the ICONICS products. There are two installations that will properly install AlarmWorX32: Genesis32 and AlarmWorX32.
- If you are installing the entire suite of Iconics 32-bit products click on **Genesis32** and refer to the **Getting Started** manual for installation instructions.
- If you are installing AlarmWorX32 as a stand alone application click on **AlarmWorX32** and follow the sequence of dialogs that follow.

B. Install from the Genesis32 Program Folders.

- Open Genesis-32/Software/Genesis32/disk1/setup.exe, OR Genesis-32/Software/AlarmWorX32/disk1/setup.exe depending on the package you are installing. As stated above, if you are installing the entire suite of products refer to the **Getting Started** manual for installation instructions. Otherwise, follow the sequence of dialogs that follow the initial setup.

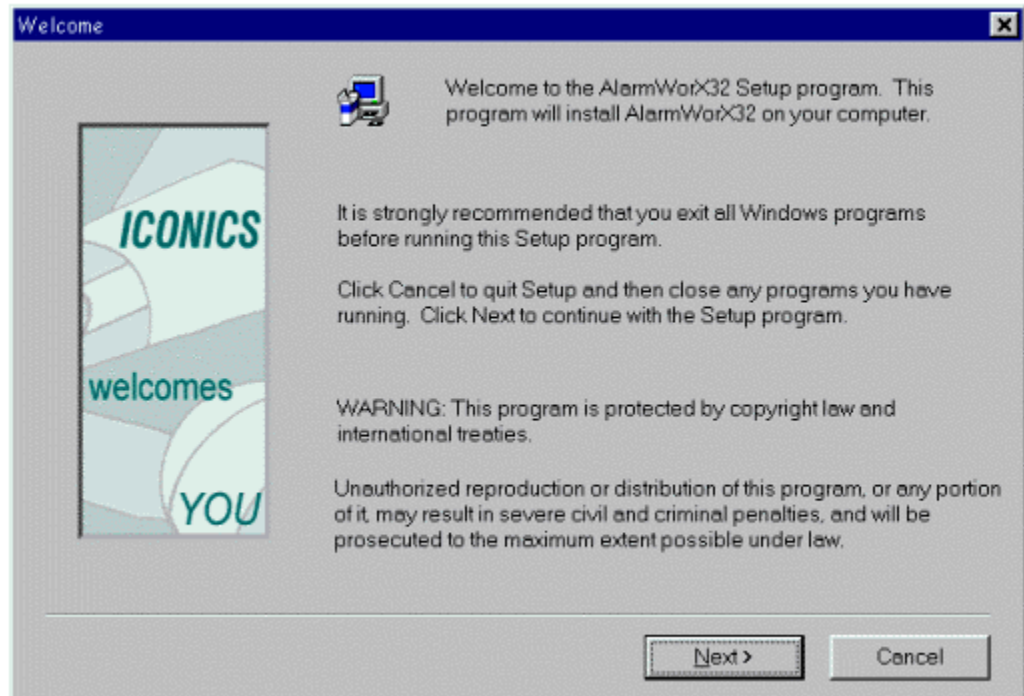
When the installation is started the following screen will appear, indicating which application you are installing.



AlarmWorX32 Installation Screen

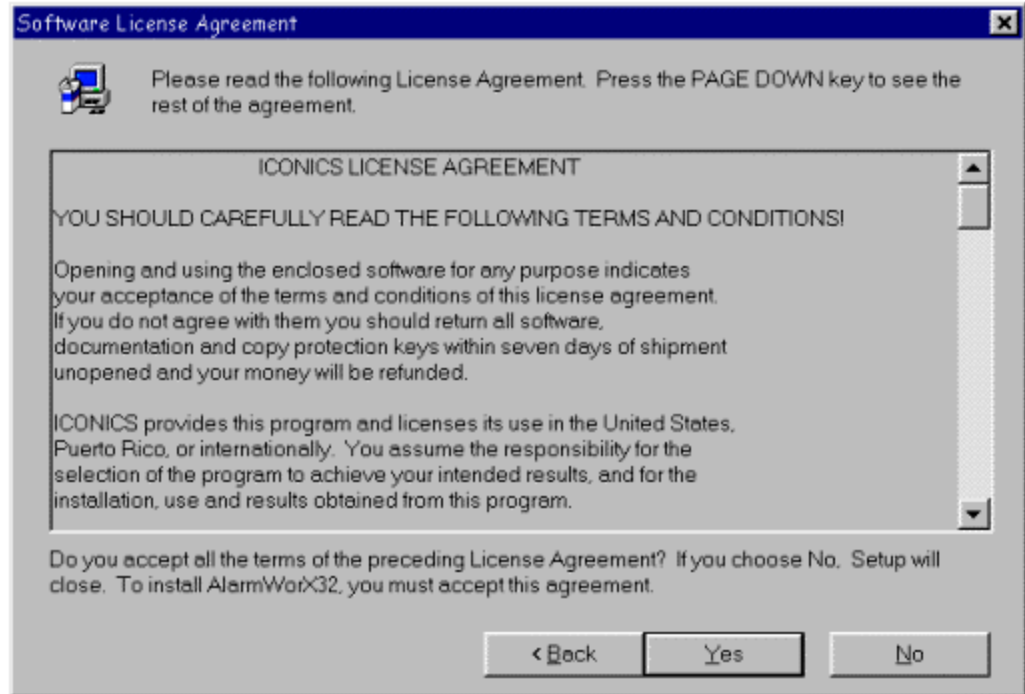
There will also be a display in the lower right corner of your screen indicating what percentage of the initial setup has been completed. When this initial setup is completed AlarmWorX32 will provide a sequence of screens providing information about Iconics and our License Agreement as well allowing the user to indicate where they want to install AlarmWorX32 and which components will be included in the installation. To navigate through these screens click the 'Next' button which appears at the bottom of each screen. If at any time you wish to cancel the installation click the 'Cancel' button.

The first screen to appear is the 'Welcome' screen, shown below:



Welcome Screen

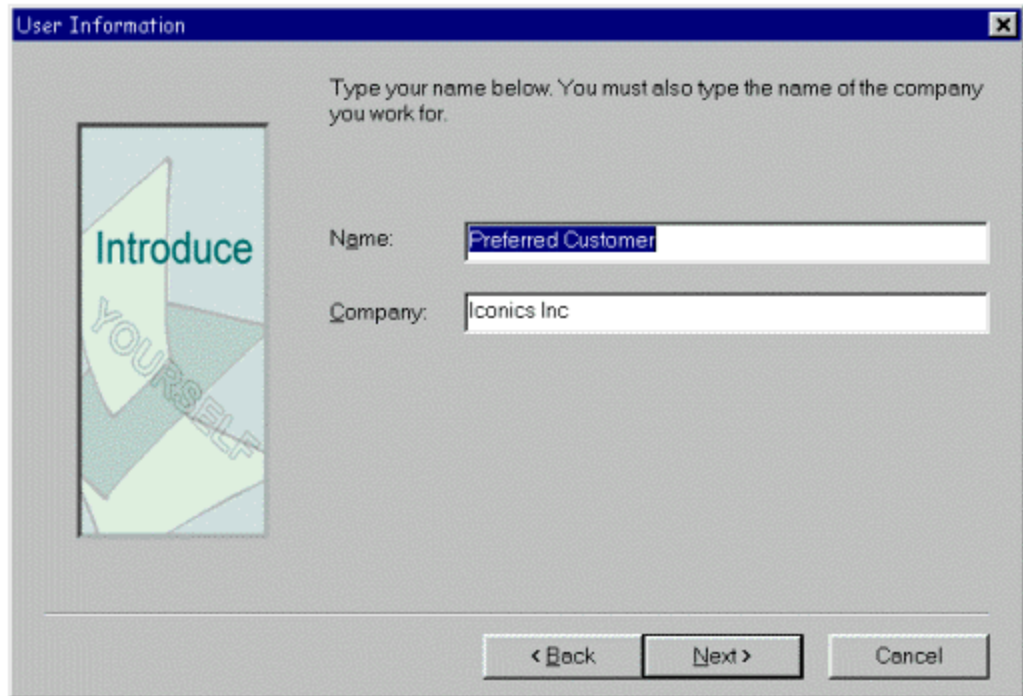
This screen provides copyright information for Iconics. The next screen that will appear is the Iconics License Agreement.



License Agreement

It is extremely important that you read and fully understand this agreement. Clicking the 'Yes' button indicates that you have read the agreement and fully understand its contents. Clicking 'No' will abort the installation.

Next you will be asked to enter your User Information.



The 'User Information' dialog box has a title bar with the text 'User Information' and a close button. On the left, there is a graphic with the text 'Introduce YOURSELF' in a stylized font. The main area contains the instruction: 'Type your name below. You must also type the name of the company you work for.' Below this, there are two text input fields. The first is labeled 'Name:' and contains the text 'Preferred Customer'. The second is labeled 'Company:' and contains the text 'Iconics Inc'. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

User Information

Type your name below. You must also type the name of the company you work for.

Introduce YOURSELF

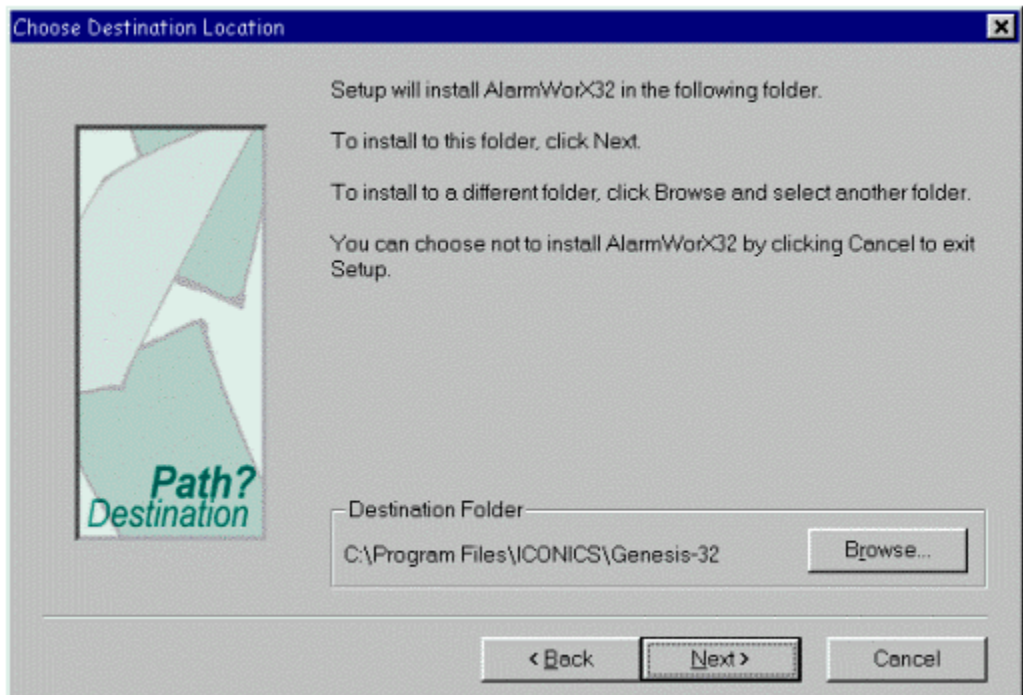
Name: Preferred Customer

Company: Iconics Inc

< Back Next > Cancel

User Information Screen

This information is necessary for the registration of the software. When you have entered the proper Name and Company name click 'Next' to continue on to the Choose Destination Location screen.



The 'Choose Destination Location' dialog box has a title bar with the text 'Choose Destination Location' and a close button. On the left, there is a graphic with the text 'Path? Destination' in a stylized font. The main area contains the following text: 'Setup will install AlarmWorX32 in the following folder.', 'To install to this folder, click Next.', 'To install to a different folder, click Browse and select another folder.', and 'You can choose not to install AlarmWorX32 by clicking Cancel to exit Setup.' Below this text, there is a text input field labeled 'Destination Folder' containing the path 'C:\Program Files\ICONICS\Genesis-32'. To the right of this field is a 'Browse...' button. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Choose Destination Location

Setup will install AlarmWorX32 in the following folder.

To install to this folder, click Next.

To install to a different folder, click Browse and select another folder.

You can choose not to install AlarmWorX32 by clicking Cancel to exit Setup.

Path? Destination

Destination Folder

C:\Program Files\ICONICS\Genesis-32

Browse...

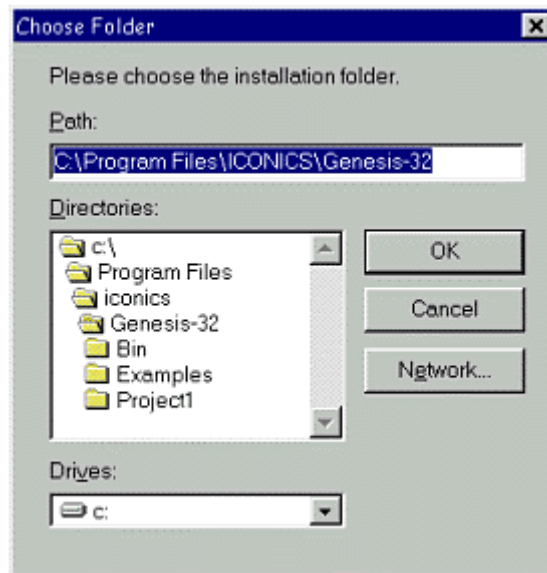
< Back Next > Cancel

Choose Destination Location Screen

Using this screen, the user can choose where they want to install AlarmWorX32. The default folder for installation is:

C:\Program Files\ICONICS\Genesis-32

To change the installation folder click the 'Browse' button and the following dialog will appear:

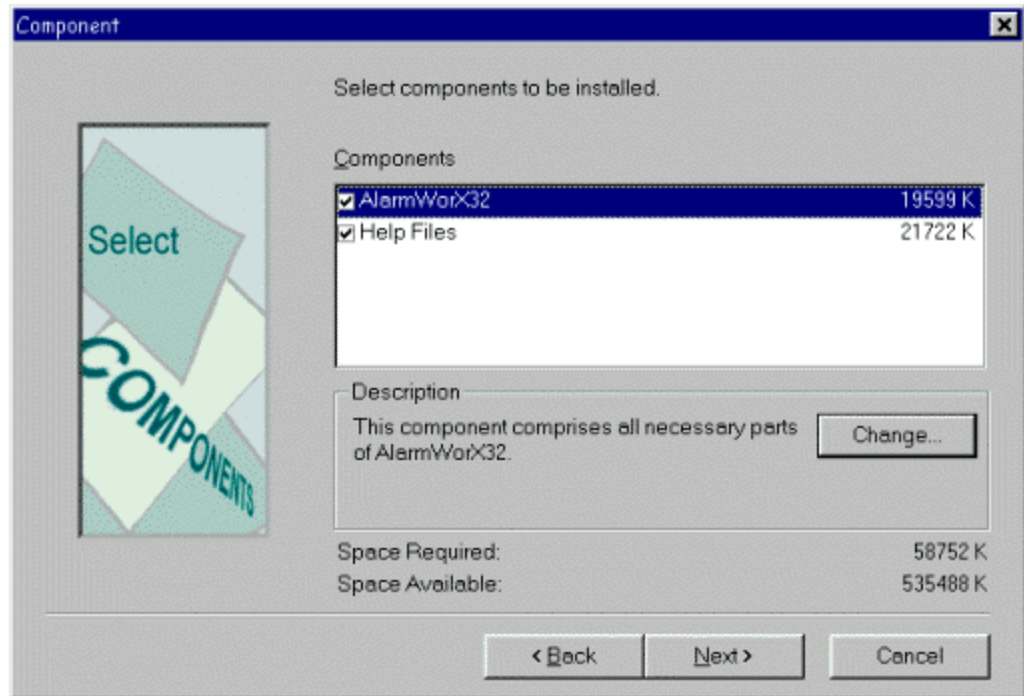


Choose Folder Dialog

The user can either type in the path where they want to install AlarmWorX32 or they can use the Directories and Drives menus to select the installation location. It is also possible to install in on the network using the 'Network' button. To navigate through the Directories menu simply double click on a folder to open or close it.

Note: It is possible to add a new folder to a certain directory. To do so select the directory using the Directories menu and in the Path field type 'NewFolder' at the end of the Path.

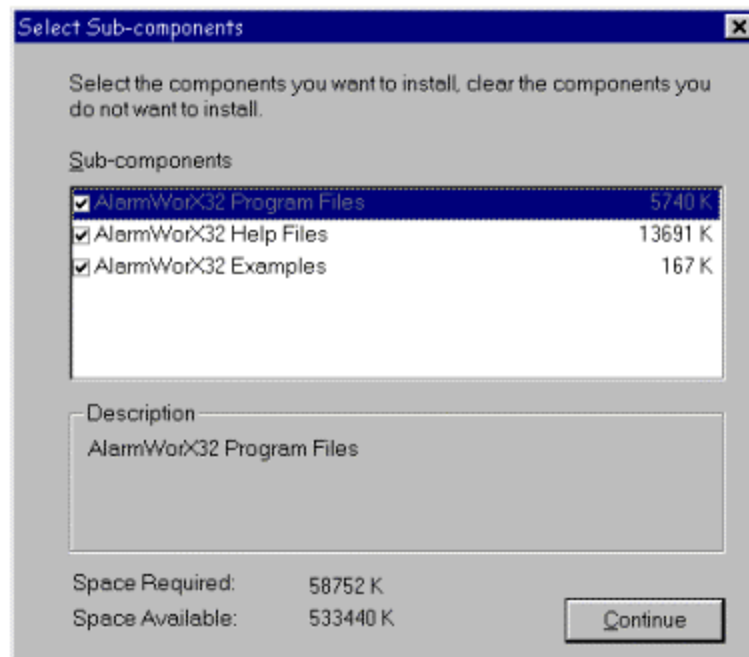
Once the user has selected the desired folder for installation, click 'Next' and the Components screen will appear.



Components Screen

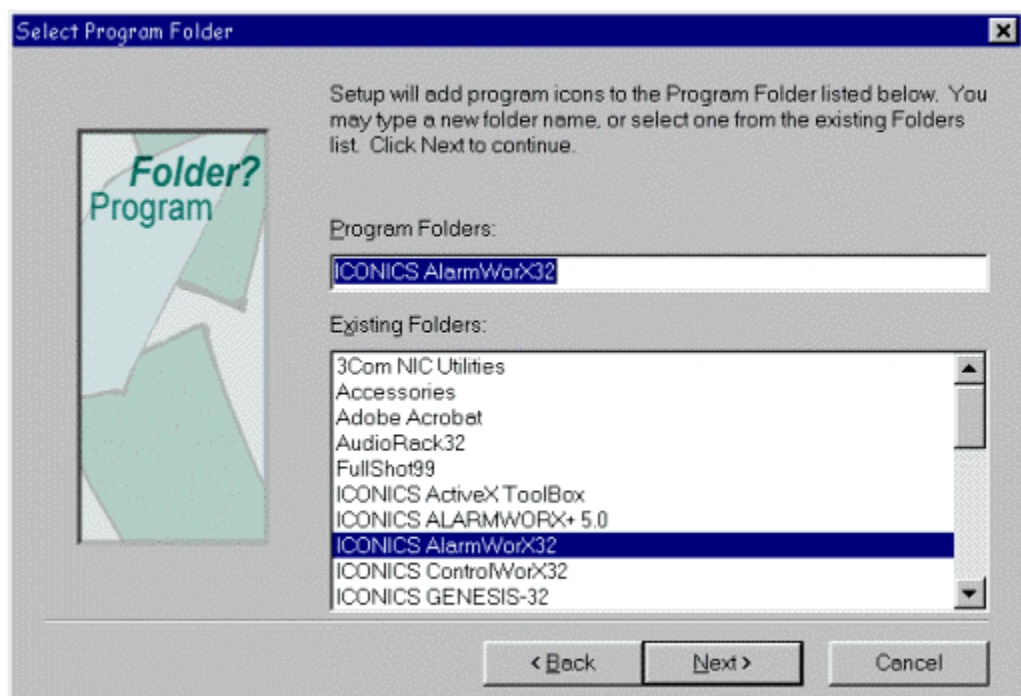
Note: To select or de-select a component click the corresponding check box.

The Components screen allows the user to choose which components they wish to install. This is helpful if they are operating on a machine that has limited space available. Clicking the 'Change' button brings up the Sub-Components screen, which gives the user even more versatility in the installation. It is important to note that if help files are not installed, online help will be unavailable.



Select Sub-Components Screen

Similar to the Components screen, the Select Sub-Components screen lists all items that comprise the component that was highlighted on the previous screen. This gives the user the ability to customize the installation. Select or de-select the sub-components and click 'Continue' to return to the Components screen. Clicking 'Next' on the Components screen will bring up the Select Program Folder screen as shown below.



Select Program Folder screen

The Select Program Folder screen allows the user to select the name of the folder they want AlarmWorX32 to appear in on the **Start | Programs** | directory. They can either choose an existing folder or type in a new folder name. By default AlarmWorX32 installation creates a folder called ICONICS AlarmWorX32.

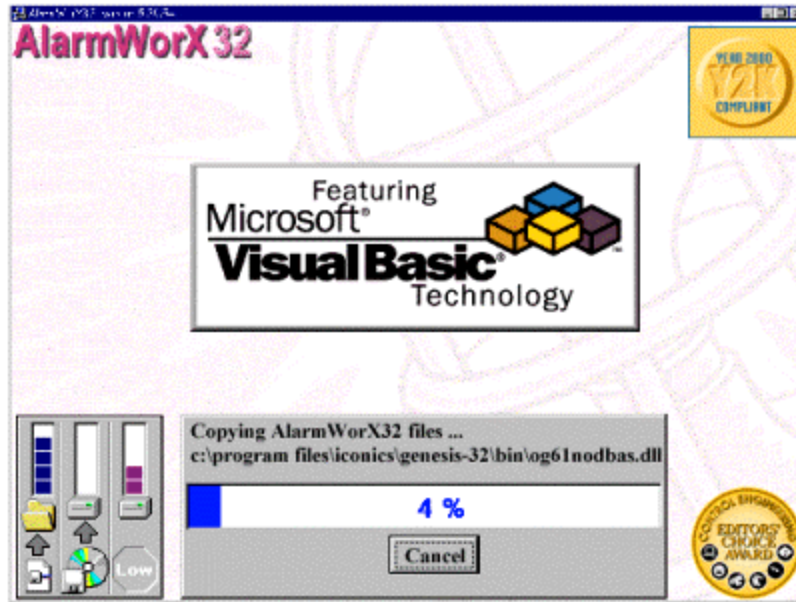
The next screen is the Security Password screen.



Security Password Screen

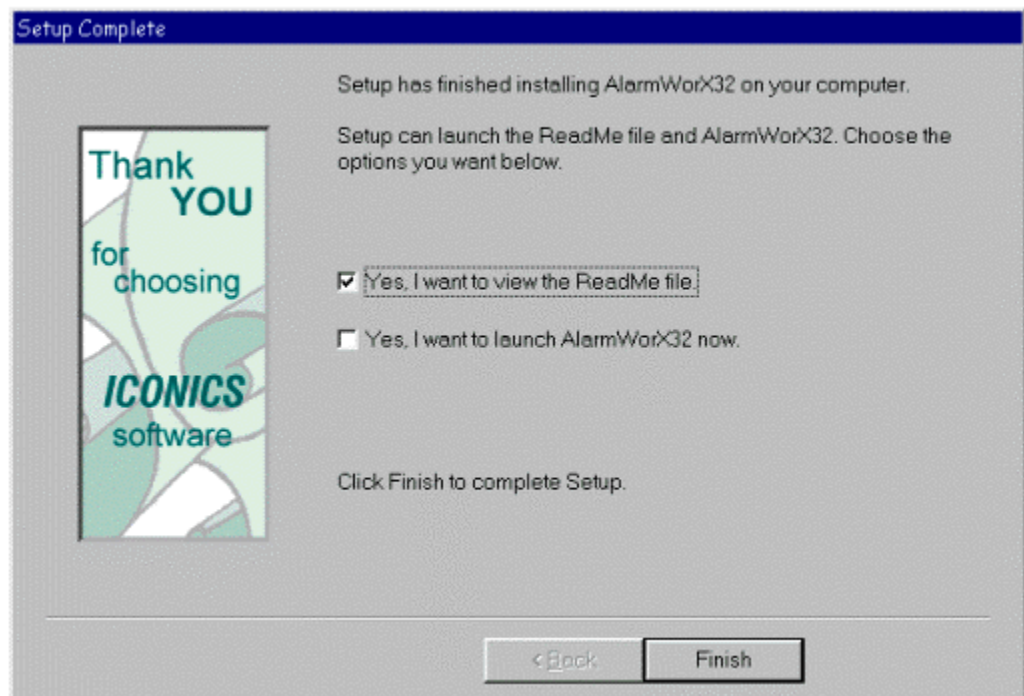
In this screen the user will choose a default security password. It is important that they choose a word which is relatively easy for them to remember and that the password is recorded somewhere so that they can easily find it should they forget what it is. The reasoning behind this is that this password is necessary when setting up the security for all Iconics products on the Iconics Security Server. The administrator password can be changed at anytime using the Iconics Security Server. After selecting a default security password, click 'Next' to start the actual installation of AlarmWorX32.

Your system will now complete the installation and will self-register several files so that the software will run correctly. While the installation is taking place the following screen will appear.



AlarmWorX32 Is Installing

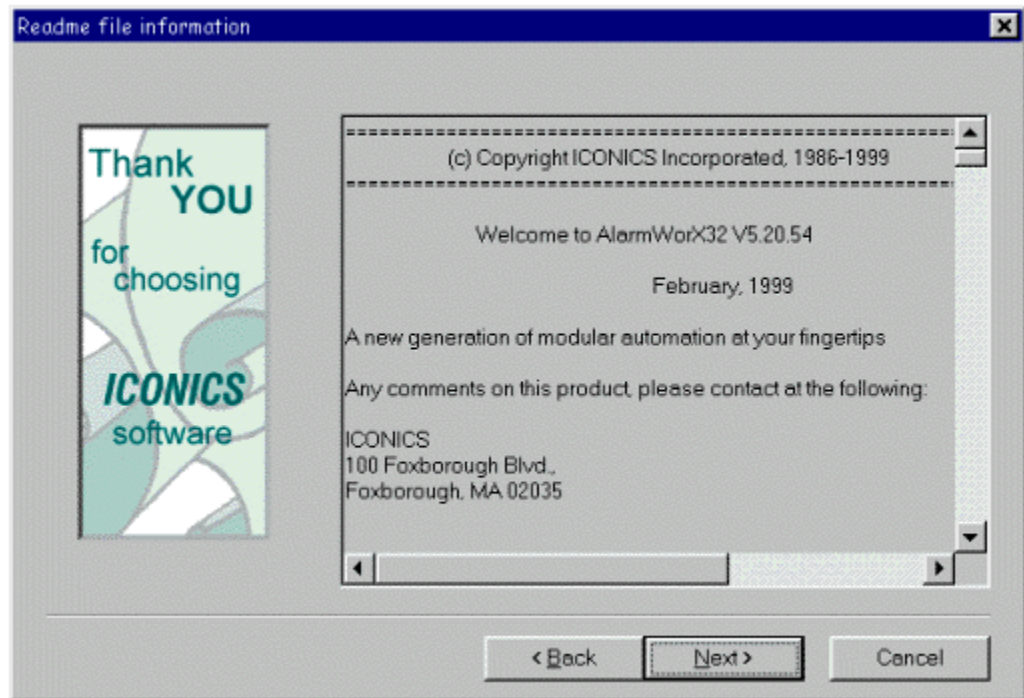
After AlarmWorX32 has completed installation the Setup Complete screen will appear as shown below.



Setup Complete

If this dialog appears it means that the installation was successful and AlarmWorX32 is ready for use on your PC. This screen also gives the user the option of opening the ReadMe file. If this option is selected

then the following screen will appear, displaying the ReadMe file for AlarmWorX32. The ReadMe contains pertinent information about the product and Application Notes that were added after the print of this manual.

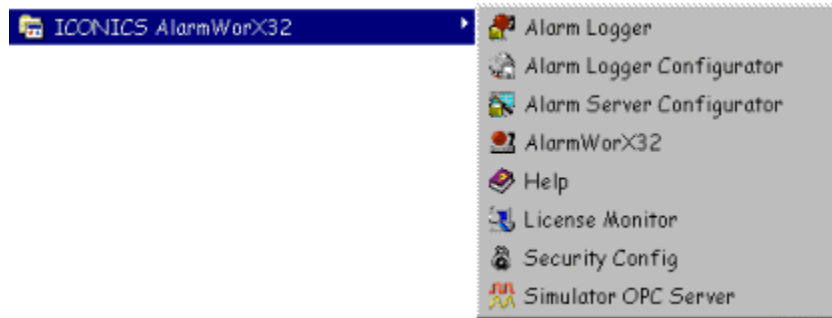


AlarmWorX32 ReadMe File

After reading over the ReadMe file click the 'Next' button and the installation will be completed.

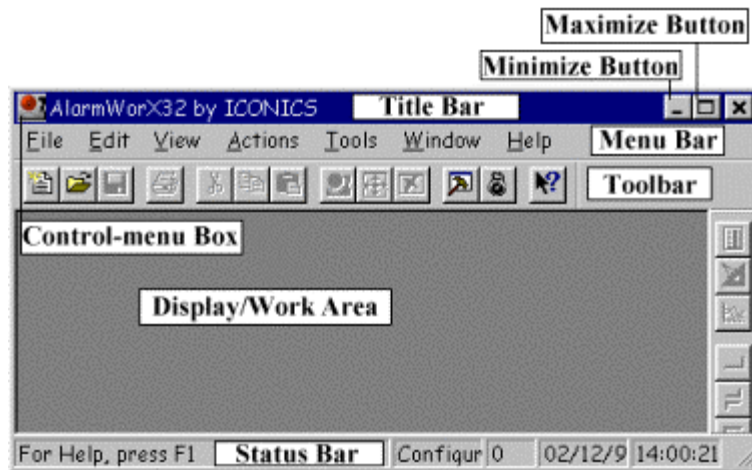
Starting AlarmWorX32 Container

To start AlarmWorX32 Container, double-click on the **AlarmWorX32** menu item, shown below (located in the ICONICS AlarmWorX32 program group by default).



AlarmWorX32 Menu

If you have not installed an ICONICS Software License you will only be able to run AlarmWorX32 in a 2 hour Demo mode. If you have a software license and do not wish to run in Demo mode, make sure that the license is properly installed. Once you start, a blank window will appear as shown below:



Blank Window

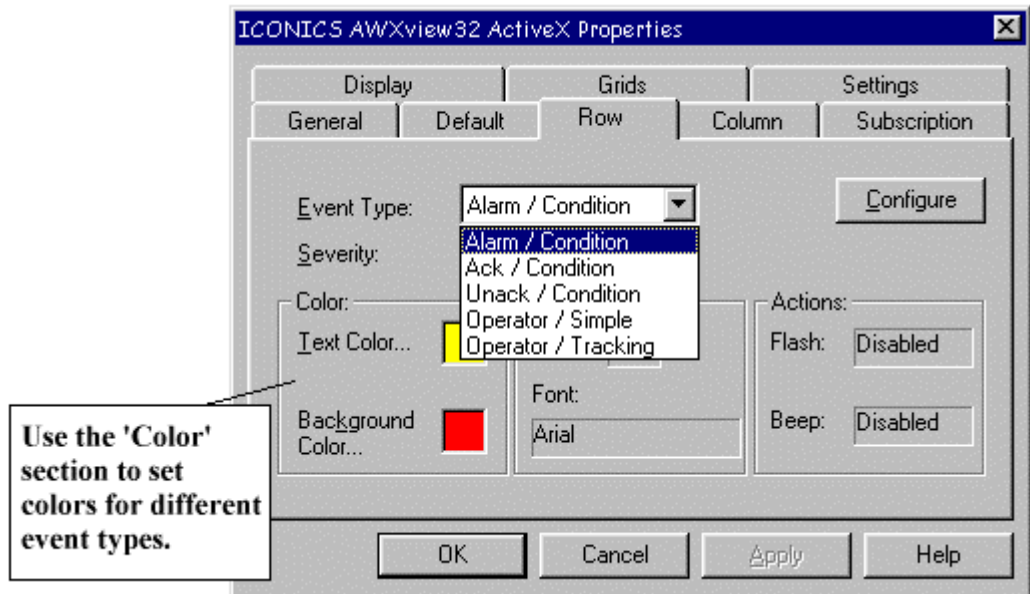
AlarmWorX32 Container Screen Components

Title Bar	Displays the name of the application and the name of the current screen displayed. To reposition the screen, click the title bar with the left mouse button and drag it to the desired location.
-----------	--

Control-menu Box	Displays the Windows Control menu. Refer to your windows documentation for more information about the Control menu.
Minimize Button	Reduces the window to an icon.
Maximize Button	Enlarges the active application window to fill the entire screen.
Menu Bar	Contains the options File, Window, and Help through which you can access the AlarmWorX32 Container features, such as printing and saving to a file.
Toolbar	Contains toolbar buttons for certain useful menu items.
Display/Work Area	Displays the various AlarmWorX32 Container screens as you select them. The information shown here varies depending upon the screen you select.
Status Bar	Displays context-sensitive help, display mode, and the current time and date.

Refer to the Screen Features chapter for a complete description of the menu bar, toolbar, and status bar functions.

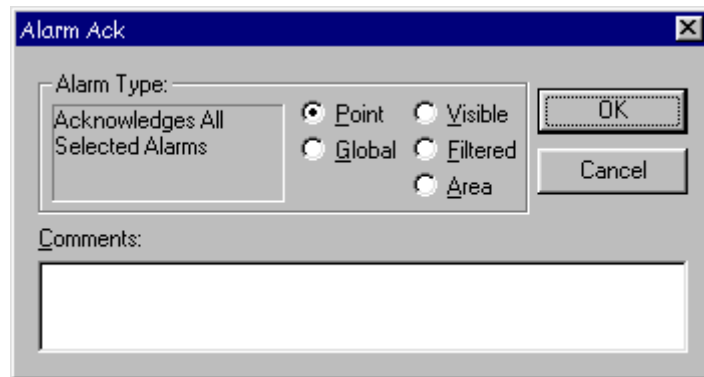
- From the toolbar select **File – Open** and use the browse functionality to get to **...installed directory\Genesis-32\Examples\AWX32 Examples**. From a standard installation this directory is **...Program Files\Iconics\Genesis-32\Project1**.
- Select AlarmSample.a32 and press the Open Button. This will open a configured Alarm Viewer file.
- From the container menu select **Action – Application – Runtime (Ctrl + M)**. At this point the container will enter runtime mode and alarms will start posting in the viewer. Based on the configuration settings in the file AlarmExample.a32, the Alarm Viewer will start the AlarmWorx32 Server. The AlarmWorx32 Server will start the Iconics.OPC.Simulator which provides simulated opc data. Within the ActiveX Properties dialog it is possible to configure column and row settings, set filtering, and configure sort format. In the example file the soft yellow background is the default background color which is set for an UnAck/Condition signal, while the red background color is set for an Alarm/Condition signal. These colors are set in the Row tab by changing the event type and setting different colors for specific event types.



Configuring Alarm Type Display

Additionally, AlarmWorX32 offers the following functionality in runtime, while right clicking on the display:

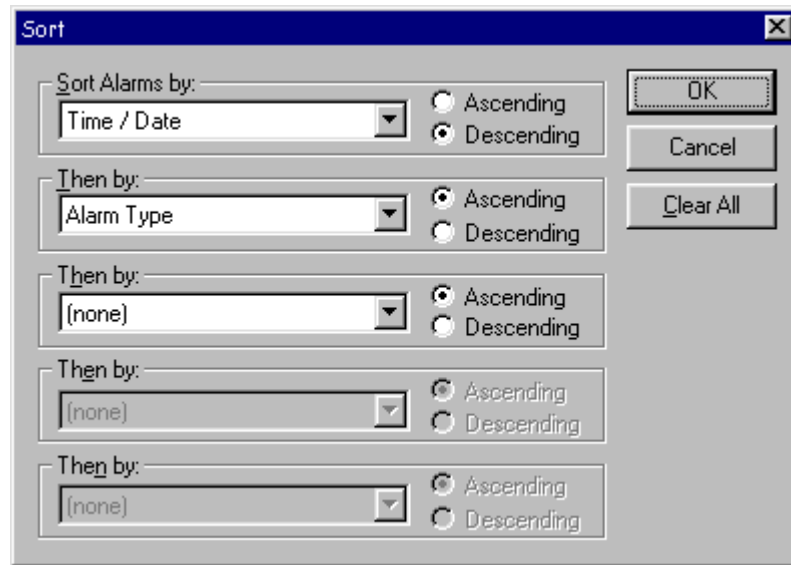
- Ack – When choosing 'Ack' from the drop down menu the following dialog will appear:



Alarm Acknowledgement Configuration

This dialog allows the user to choose which type of alarm acknowledgement they wish to enact: Point, Global, Visible, Filtered or Area. The user can see what the functionality of each type of acknowledgement is by looking in the information box to the left of the ack choices. To connect a comment to the acknowledgement type it in the 'Comments' box at the bottom of the dialog. When the user has selected the type of acknowledgement that they desire, click 'OK' to enact the acknowledgement or 'Cancel' to leave the alarms as they are. Once an alarm is acknowledged its display colors will change to those configured by the user in the ActiveX Properties dialog under the Row tab (see above for further explanation).

- Sort – When selected the Sort dialog will appear:



Sort Dialog

Using the 'Sort' dialog the user can determine what criteria they want AlarmWorX32 to use when determining the order in which the alarm signals should be displayed. It is possible to set up to five different criteria for sorting and the user can change the order by simply clicking down arrow and selecting an available column to sort by. The columns that are available are determined by the choices made by the user in the 'Column' tab of the ActiveX Properties dialog. If a user has added an additional attribute under a certain subscription, this attribute will become available as a sorting criterion.

- ToolTips – When selected the user will be given the option of selecting which type of ToolTip they want to have displayed during runtime: None, Context, Help, Comment or Alarm Line. Depending on the choices made on the 'Settings' tab in the ActiveX Properties dialog some of these choices may not be available.
- Language – When selected the 'Select Language' dialog appears as shown below:



Select Language Dialog

Select the desired language and click 'OK'. If the selected language is not supported by AlarmWorX32 a dialog will appear informing the user of this fact.

- VB Event - This option allows the user to trigger an automation event. To use this event requires that VBA programming be tied to the user event.

To stop the system select **Action – Application – Configure** from the toolbar menu.

Display Tricks

Introduction

Color

ToolTips

Miscellaneous

Introduction

The ability to customize visual information is an important feature of HMI software. Alarm software configuration has increased in its complexity, due to the vast majority of different needs, design tastes corporate standards, and plant processes. This section will attempt to simplify some of the layout process and provide some ideas for design consideration.

Color

Outside of raw data, color is probably the single most important piece of information for an alarm display. The color configuration of the current event alarm viewer is based on three levels: default, row, and column. A judicious combination of these levels can greatly simplify configuration time. The default settings are applied when a column or row setting is not provided. If a majority of the settings require a background color of white, set the default background color to white. Row settings are used a majority of the time to provide specific colors for various alarm states. If severity is not important, it is not necessary to have multiple color configurations for severity. On the other hand, multiple color configurations for severity can be used to display sub-conditions. For example, if all High alarms are configured to be severity 700, then a color of purple for severity 700 would only show up for High alarms even though High alarms are a sub-state of the limit condition. Column settings can be used to good effect to provide constant color combinations. This is good a place to set colors for help text or to set a column apart from the rest of the display.

ToolTips

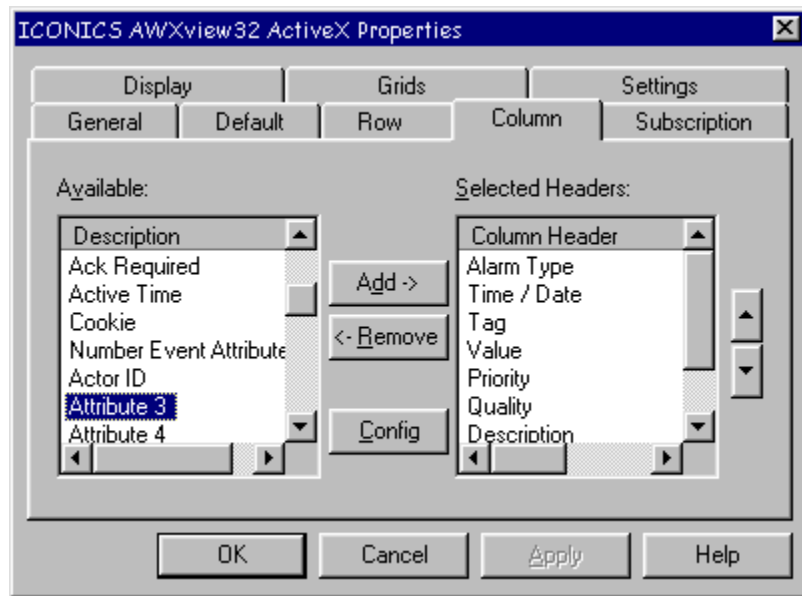
ToolTips are an easy way of providing extra information to the operator without constantly using screen resources. ToolTip configuration involves setting the visual format and the subscription.

Example

Load the **AlarmSample.a32** file found in the AlarmWorX32 Examples directory.

Note: If you are continuing on from the previous example you should already have AlarmSample.a32 open.

Bring up the property pages by either double clicking (note: only available in some ActiveX containers) on the ActiveX or right clicking and selecting the properties menu item in the popup menu. Select the columns tab.



Column Tab

Scroll down the list of available columns until Attribute 3 is visible. Add Attribute 3 to the Selected Headers list by highlighting Attribute 3 and clicking the Add button. This adds the Attribute 3 to the column list. Right click on Attribute 3 and select rename from the popup menu. Rename Attribute 3 to help2. In setting up the visual format quickly, it is faster to configure all of the columns before configuring the rest of the visual settings. To configure these settings go to the 'Settings' tab.

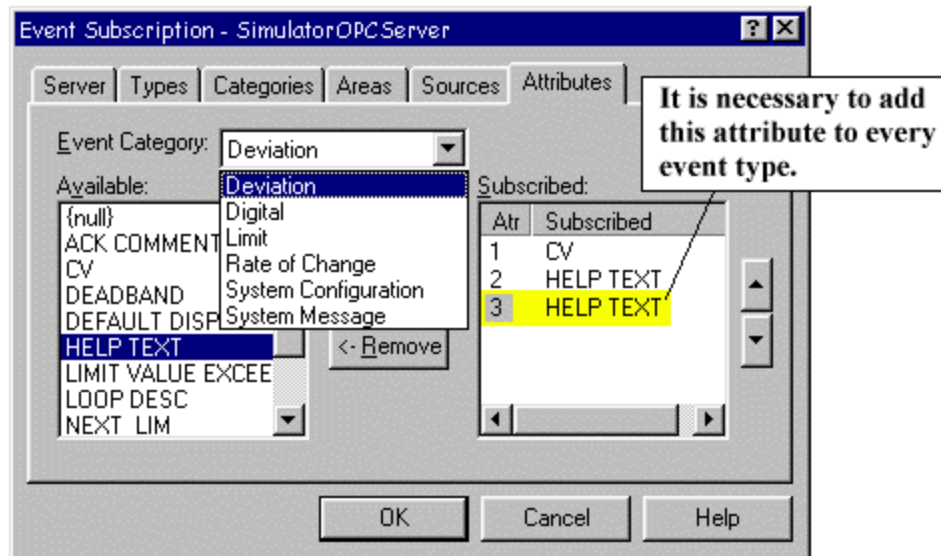


Settings Tab

In the Settings tab change the associated column for the help ToolTip to the help2 column. Do this by selecting the combo box associated with help and choose the help2 item in the drop down list. By doing this the help2 column is now associated with the help ToolTip by choosing 'Comment' in the ToolTip section.

Now that the visual side of things is set up it is necessary to change the subscription to request the additional help information. Select the subscription tab, highlight the Iconics.AlarmEventSvr subscription, and press the edit button. This will bring up the subscription dialog. Select the Attributes tab.

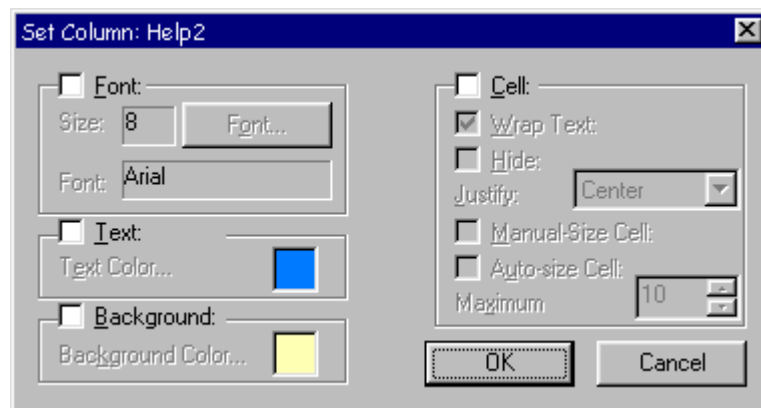
Note: As in the following figure, we can see that Current Value 'CV' and Help "HelpText" have already been subscribed for you in this example.



Attributes Tab

Add the Help Text extra attribute to the subscribed list by highlighting Help Text and pressing the Add button. This will add Help Text to the Subscribed list. Please note that order is important. Since value is currently being used as Attribute 1 and help2 is assigned to the Attribute 3 column the subscription should bring in Help Text to Atr 3. Continue this process for Limit, Digital, and Rate of Change Categories. This can be achieved by changing the Event Category item and repeating the above step. Press the Ok button on the event subscription dialog and the Ok button on the ActiveX property page. Enter Runtime and right click on any alarm. Highlight the ToolTips in the popup menu and select Help. This changes the ToolTip from Context sensitive to Help. The ToolTip information will match the information in the Help column.

As an extra step, it is possible to hide the Help2 column. Exit runtime and bring up the property page. Select the Column tab, highlight the Help column, and press the 'config' button. This brings up the column configuration dialog.



Column Configuration Dialog

Enable the Cell override feature by checking the Cell checkbox. Hide the column by checking the Hide checkbox. Press the Ok button and the Ok button on the property page. The help column will no longer be visible. When entering runtime it will still be possible to receive the help information via the ToolTips. Simply right click on an alarm line, select ToolTips, and select the help ToolTip. The default ToolTip and ToolTip width will be changed in the Settings tab of the property page.

Miscellaneous

Several features of AlarmView32 only work in conjunction with other features. For example, the wrap text feature will only be visible if the line width is two or greater. Also, it is not possible to change row widths during runtime if the row header is not enabled. When sizing columns, use the default column size to approximate the best general settings. Then use the column override to individually size any columns not exactly correct.

Subscription/Filtering

Introduction Average Filter Filtering on High Alarms

Introduction

Data is the most important piece of information displayed by an alarm viewer, however for a display to be truly effective only pertinent information is desirable. In alarm systems, filtering represents a major piece for weeding out unnecessary data. Using the OPC Alarm and Events Specification, filtering is largely achieved by creating alarm subscriptions with various servers. This effectively creates a server side filter, improving speed and reducing communication traffic. Rather than go over the information found in the OPC Alarm and Events Specification or the AlarmView32 documentation, several filtering tricks will be presented to provide a few ideas on possible filtering schemes.

Average Filter

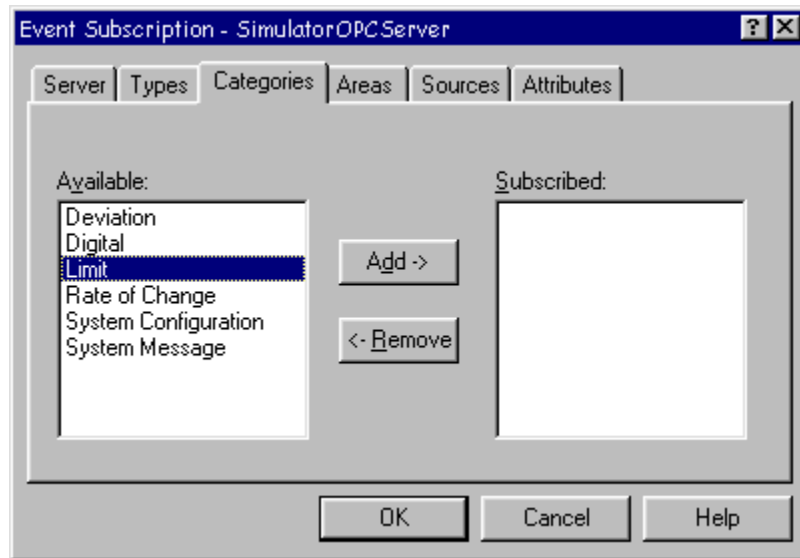
The process of setting up a subscription does more than just enable data to be passed from client to server, it also creates a server side filter. Server side filters have the advantage of reduced communication traffic between the client and servers, due to the fact that only requested messages are sent. An average filter will consist of multiple subscriptions to the same server. An example of this might be a subscription that only requests alarms from Area1 with a subscription that requests only the Rate of Change alarms.

Note: The Area can be set using the 'Area' tab (under subscription) and clicking 'browse' to find the appropriate area. Rate of Change is set on the Categories tab (under Subscription) by select Rate of Change as the event category.

This will create a filter that will send all alarms for Area1 and all Rate of Change alarms. These filters can be further refined by setting severity levels and event types. By setting low severity to 500 and enabling condition events on the Types tab, the filter can limit messages to alarms that require acknowledgment and have a severity range of 500 to 1000.

Filtering on High Alarms

Due to the open architecture of the OPC Alarm and Events specification and the use of server side filtering, some of the more specific filtering capabilities are done by client side filtering. Server side filter operates by restricting alarm messages sent to clients. This makes it more efficient than client side filtering which operates by hiding alarm messages. It is possible to only use server side configuration, but this requires filtering to be taken into account when configuring the alarm database. An example of this would be filtering on only high alarms. The first step is to make the limit condition the only requested Event Category. Select the categories tab in the Event Subscription dialog.



Categories Tab

Highlight the limit selection and press the add button. Limit will now show up in the Subscribed list. Only limit alarms are requested from the server for this subscription. It is still possible to have other subscriptions to the server that request other alarm categories. To restrict the subscription further requires some configuration work on the server side. It is possible to set alarm severity for all alarm types in the Iconics OPC alarm and event server. By setting alarms to a specific severity range it is possible to filter subconditions (high, lo, hihi, etc.) by filtering on severity. This method uses the subscription capabilities of the alarm server while taking full advantage of server side subscription (i.e. no wasted messages).

Additional Features

Using the Mouse and Keyboard Shortcut Keys Reinstalling AlarmWorX32 Container

Using the Mouse and Keyboard

The mouse is used in AlarmWorX32 to select and point to items. The mouse pointer indicates where you are in relation to the action on the screen. The pointer is usually an arrow.

In AlarmWorX32, you can use the mouse to:

- Select icons, buttons, and other screen objects
- Resize AlarmWorX32 Container screens
- Reposition windows and dialog boxes
- Scroll list boxes
- Display context-sensitive help in the status bar for screen objects, such as icons, buttons, and parameter fields

The table lists the various mouse functions and descriptions with which you need to be familiar.

Mouse Functions

Function	Description
Click	Position the mouse pointer over the object and press the left mouse button once.
Double-click	Position the mouse pointer over the object and press the left button twice in quick succession.
Drag / Drop	Click on the object and, with the button still depressed, move the object to the desired location and release the button.
Select	Click once on the object with the left mouse button, highlighting it (enclosing it in a rectangle with a dotted line rule or in reverse color).
Move pointer over	Place the mouse pointer on top of a object, such as an icon or button, to display context-sensitive help for that object in the status bar.

Shortcut Keys

The keyboard is used for entering information into dialog boxes and invoking selected menu options. This table presents the available shortcut keys and their related functions (English version only).

Shortcut Keys

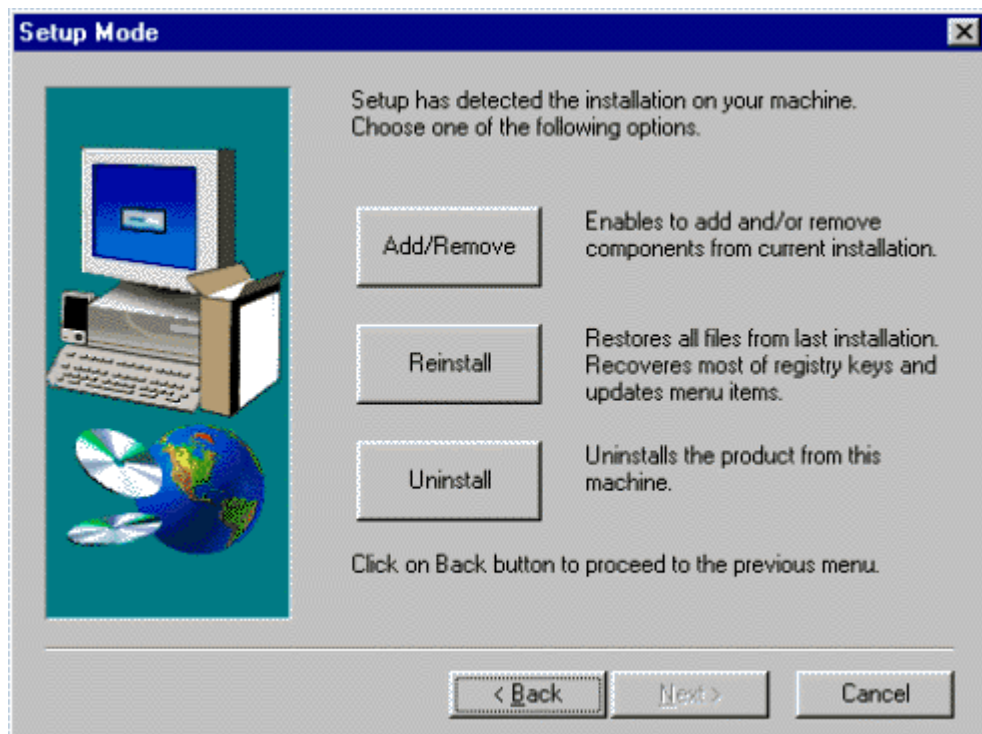
Key	Function
ALT+F	Displays the File menu options available for the displayed screen.
ALT+H	Displays the available Help menu options.
ALT+T	Displays the Tool menu options available for the displayed screen.
ALT+W	Displays the Window menu, which contains a list of available AlarmWorX32 Container screens.
ALT+V	Displays the View menu options available for the displayed screen.
ALT+A	Displays the Actions menu options available for the displayed screen.
ALT+E	Displays the Edit menu options available for the displayed screen.
ALT+F8	Starts the Macros dialog.
ALT+F11	Starts the Visual Basic editor.
CTRL+P	Prints the current screen's information.
CTRL+N	Opens a new Alarm window.
CTRL+O	Opens an existing Alarm window.
CTRL+S	Saves the currently selected Alarm window.
CTRL+T	Displays the Main Toolbar.
CTRL+U	Displays the Status Bar.

CTRL+F	Sizes the Alarm window to the display.
CTRL+G	Tiles all the open Alarm windows in the display.
CTRL+H	Cascades all open Alarm windows in the display.
CTRL+A	Animates the display.
CTRL+Shift+A	De-Animates the display.
CTRL+M	Toggles the display between Runtime and Configuration mode.
Del	Deletes an object from the display.

F12	Inserts a new object in the display.
F9	Launches Configurator
F2	Displays Properties Window for object inserted in the display
F3	Brings up Display Preferences Dialog for object inserted in display
F4	Displays Application Preferences Dialog for object inserted in display
F6	Enables Security Login
F7	Enables Security Configuration
F1	Displays Help Topics

Reinstalling AlarmWorX32 Container

If you try to reinstall AlarmWorX32 the following dialog will appear:



Setup Dialog When Reinstalling

- If you are having problems with, or are missing certain components you can click on the 'Add/Remove' button to add or remove these components.

- To update your version of AlarmWorX click on the 'Reinstall' button and AlarmWorX32 Container will update your system, restoring most of your old files.

If you wish to uninstall your current version of AlarmWorX32 click the 'Uninstall' button. You will lose all of your old files if you choose this option.

AlarmWorX32 Container Application

This ActiveX container application integrates the use of the following components.

- **Alarm Logger Configurator**
- **Alarm Viewer ActiveX**
- **Alarm Reporting/Analysis ActiveX**
- **ICONICS OPC Alarm Server**
- **Alarm Server Configurator**
- **AlarmWorx32 Container**

This is a multiple document interface (mdi) application. It provides a mdi ActiveX container which integrates the various ActiveX alarm components and configuration tools into one user friendly tool. Menus and toolbar items are provided to allow easy access to all configuration and runtime features of all of the AWX32 components. It also adds VBA (Visual Basic for Applications) scripting support.

About the AlarmWorX32 Container

Screen Features: File

Status Bar and Toolbar

ActiveX Toolbar

File Menu

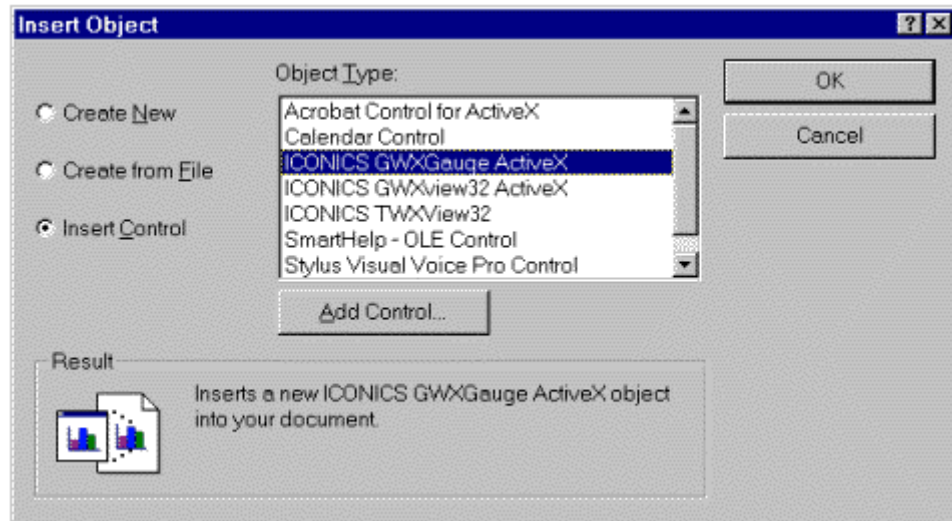
Printing

Edit Menu

ActiveX Object Pop-up Menu

About the AlarmWorX32 Container

As discussed previously, a container is an application that houses other objects. AlarmWorX32 works as a container by housing ActiveX objects which are listed in the 'Edit - Insert Object' menu item.



Insert Object Dialog

Clicking the 'Add Control' button brings up a browse dialog and allows the user to select an existing ActiveX object from the list of ActiveXs installed on your system.

Screen Features: File

The menu bar varies for most AlarmWorX32 screens and contains the following menus:

- File
- Edit
- View
- Actions
- Tools
- Window
- Help

This chapter explains the various features available from the menu bar, toolbar, and status bar that allow you to manage your database records, and use the AlarmWorX32 container easily and efficiently. This section essentially explains how you can edit the object that is inserted into the AlarmWorx32 container. It discusses the File and Edit menu items that enable the user to modify the appearance of the display and change the integral properties of the ActiveX object that is housed by the container.

Note: The File functions vary according to the screen displayed. (This is documented where appropriate.)

Status Bar and Toolbar

The status bar and toolbar are objects within each window. The information contained in the status bar varies depending upon the current screen displayed and the current system configuration and activity.

Status Bar

The status bar (located at the bottom of each AlarmWorX32 screen) contains areas for displaying help information, display mode, the current time, and the current date. The following displays a status bar view from AlarmWorX32.



Status Bar

The following table explains the various information and objects that may display in the status bar.

Status Bar Parameters

Parameter	Description
Context-sensitiveHelp area	Displays information about the screen object over which the mouse pointer is currently positioned.
Current Mode	Runtime or Configure (depending on the current mode)
Number of Object(s)	Displays the number of open ActiveXs.
Current time	Displays the current time. Click on this field to display the amount of disk space available. This field turns red if your available disk space drops below 100k.
Current date	Indicates current date.

Toolbar

The toolbar (located below the menu bar) contains icons representing the different functions available in AlarmWorX32. The following shows an example toolbar.



AlarmWorX32 Toolbar

As you move your mouse pointer over each icon, the status bar (located at the bottom of the screen) displays context-sensitive help indicating the function of the icon.

ActiveX Toolbar

Additionally, an ActiveX Toolbar has been added to the container. This toolbar provides the following ActiveX's: Gauge, GraphWorX32, TrendWorX32, Event32, Periodic, SCR Alarm, Button, Toggle Button, Check Box, Option Button, Label, Text Box, List Box, Combo Box and Scroll Box. Please note: if an ActiveX is not installed on the current PC it will not be found on the ActiveX Toolbar.

File Menu

The following functions are available from the File menu:

- New - CTRL + N
- Open - CTRL + O
- Close
- Save to File
- Print - CTRL+ P
- Print Preview
- Exit

If an Alarm window is not open the Close, Save and Print options will be disabled.

Printing

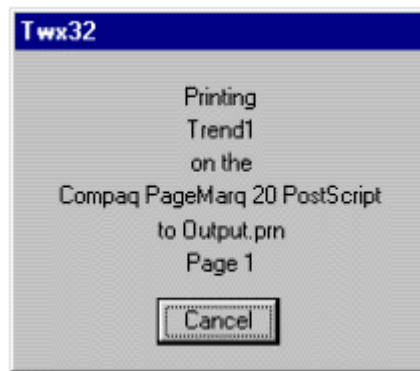
Using AlarmWorX32, you can perform any of the following printer functions:

- Print the Current Window
- Preview the screen's information before printing
- Print only the Help instructions for the currently selected alarm

To configure a printer, you need to define the appropriate parameters in the Microsoft Windows Control Panel. Refer to your Windows documentation for complete information.

To Print Screen Information

Select **Print** from File menu, or press **CTRL+P**. AlarmWorX32 prints all records contained within the scroll window on the screen. When the information is sent to the print queue, the Printing status box shown below appears.



Printing Status Box

The Printing status box indicates the number of records printed and the page that is printing.

To Cancel Printing

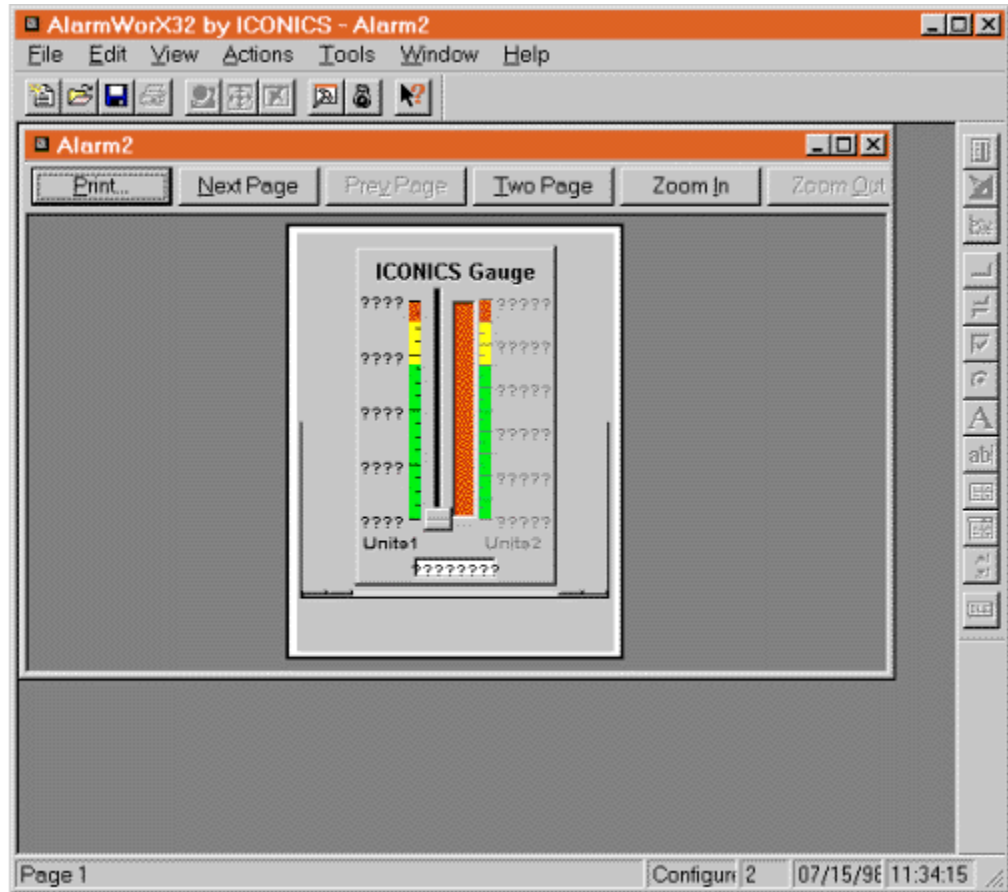
Click the **Cancel Printing** button in the Printing status box to cancel printing. Printing halts immediately and the print job is removed from the print queue.

Previewing Your Print-out

You can initiate a print preview from any of the four screens from which you can print.

To Preview Screen Information Before Printing

1. Select Print Preview from the File menu. The Print Preview screen displays, as shown below.



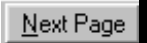
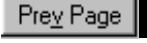
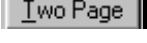
Print Preview

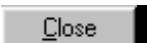
This figure shows a print preview invoked from the Trend window for a time plot.

Print Preview Parameters

Parameter	Description
Title Bar	Indicates the file currently displayed.

Page Selection Buttons

	Print button	Sends the all the pages in the current preview to the Windows Print Manager for printing.
	Next Page button	Displays the next page in a multiple page print preview.
	Previous Page button	Displays the page immediately before the current page in a multiple page print preview.
	Two Page button	Displays two pages at a time in a multi-page print preview.

	Zoom In button	Changes view of the Print Preview screen from normal view.
	Zoom Out button	Changes view of the Print Preview screen to full page view. Full Page view allows you to see how the entire page looks upon printing.
	Close button	Exits the Print Preview view and returns to an editable/Normal view.

2. Use the buttons explained in the previous table to view the preview information before you print.

3. Click the **Print** button to display the Print dialog box.

The Print dialog box indicates the configured printer to which this information prints. Refer to your Windows and printer documentation for complete information about configuring a printer.

4. Select the button to indicate which information should be printed (either all pages or a specified selection range) and click the **OK** button or press **ENTER**. The information prints to the configured printer.
5. Double-click on the Control-menu box located in the upper-left corner of the screen, or click the **Close** button to close the print preview screen.

Edit Menu

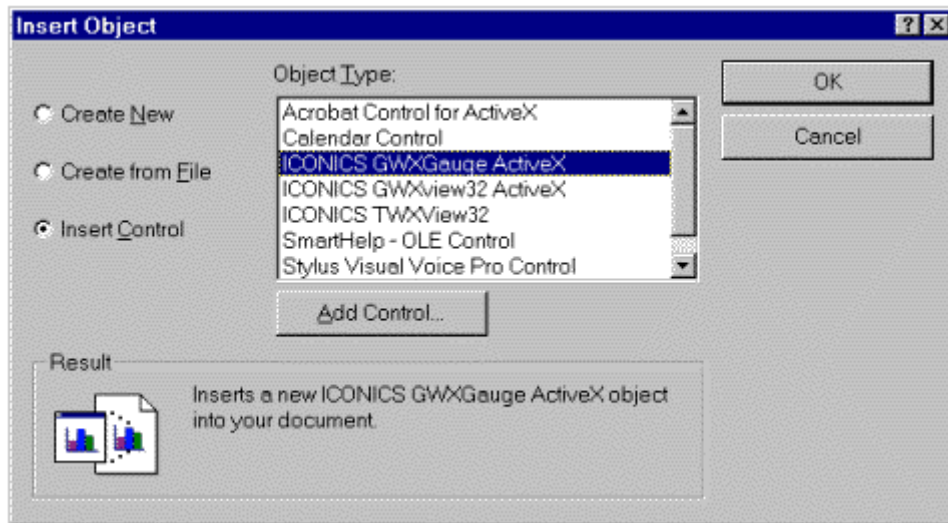
The Edit Menu contains the following functions, which are discussed in this section:

- Cut
- Copy
- Paste
- Paste Special
- Delete Object
- Insert New Object
- Links
- Object
 - * Properties
 - * Convert

Use the 'Insert New Object' items to add objects to the current window. Alternatively integrated Iconics ActiveX controls can be inserted by selecting the corresponding toolbar button. To configure the properties for an object either double click on that object or right click and select the 'AWXView32 ActiveX Object' - 'Properties' item from the popup menu. The Properties dialog is described further in the AlarmWorX32 Viewer section.

Insert Object

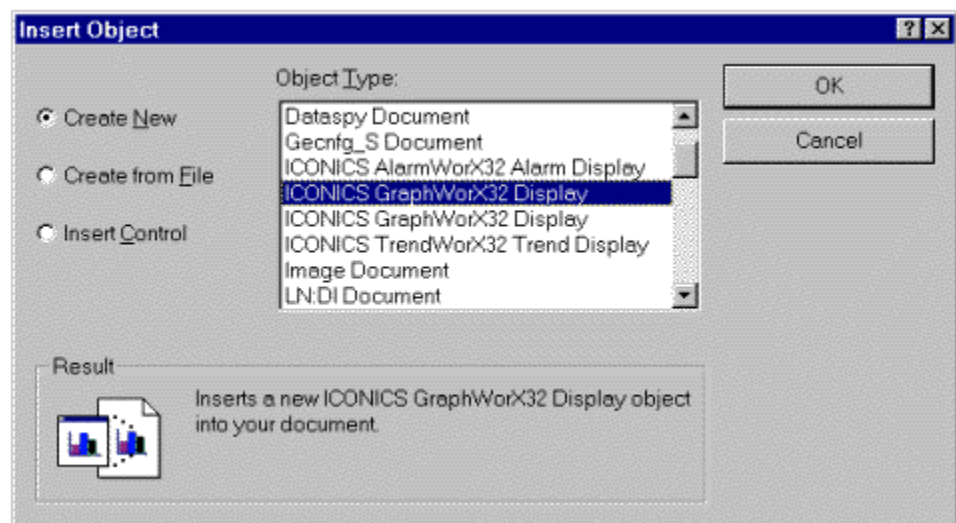
Selecting the insert Object menu item under the Edit menu displays the following dialog:



Insert Object Dialog

Create New

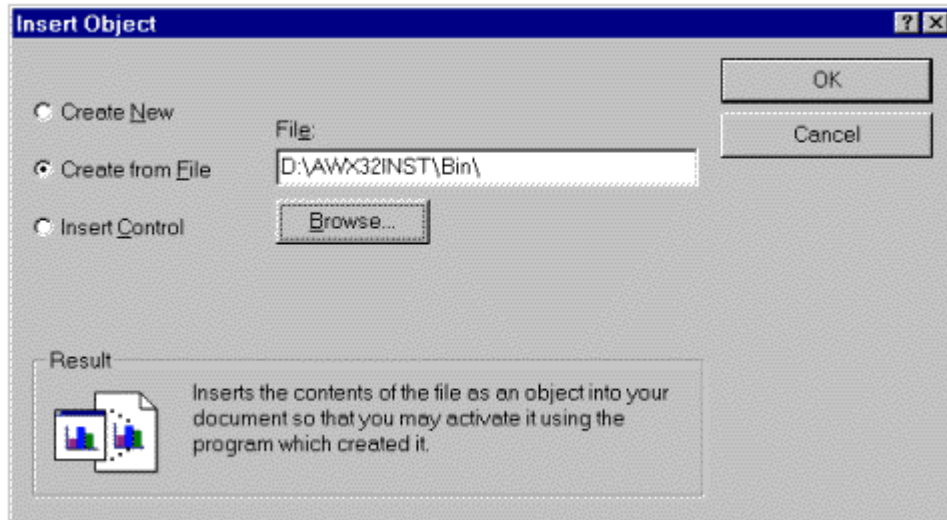
This option allows the user to create a new object and insert the object in the container. Selecting this tab brings up the following dialog, which offers a list of object types to choose from as shown below:



Create New Object Tab

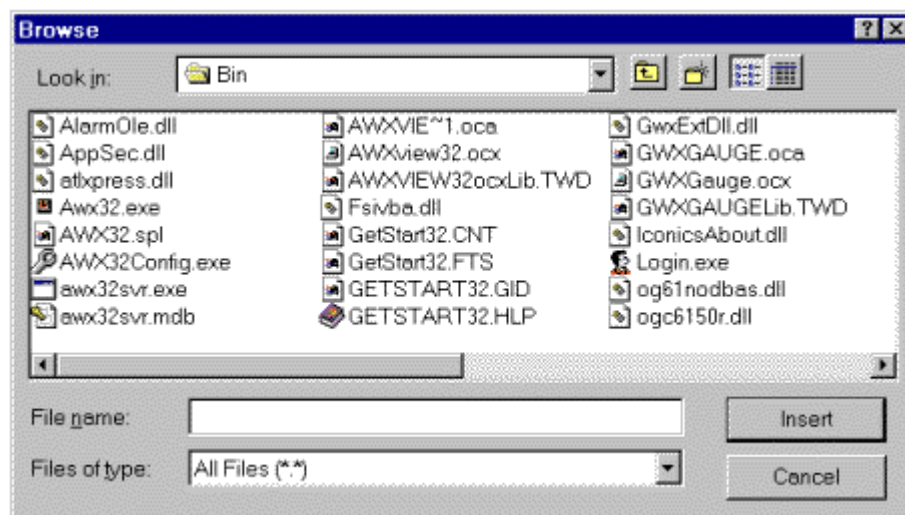
Create from File

This Tab provides the option to insert an object into the container that is created from an existing file. As shown in the following dialog the user has the option to browse available directories and search.



Insert Object Dialog - Create From File Tab

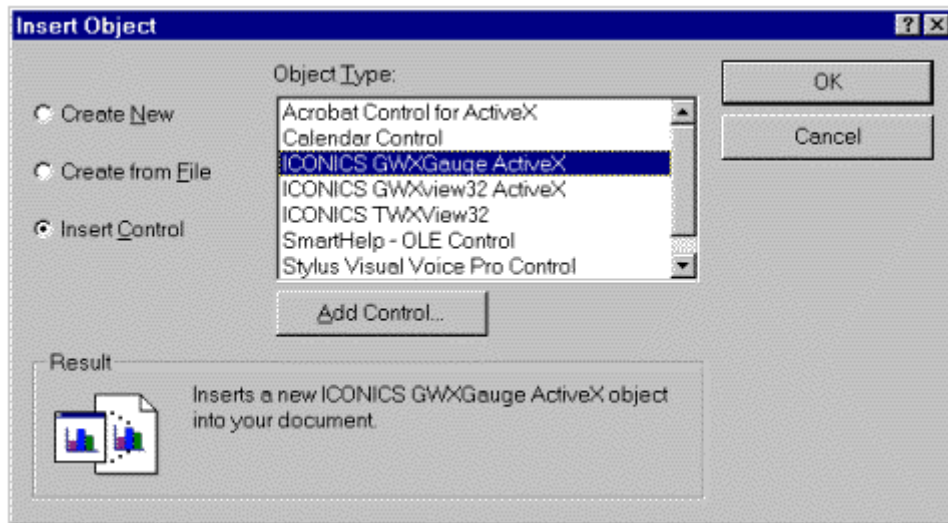
Clicking the 'Create from File' button brings up a browse dialog and allows the user to select an existing ActiveX object from the list of ActiveXs installed on his/her system.



Browse Window

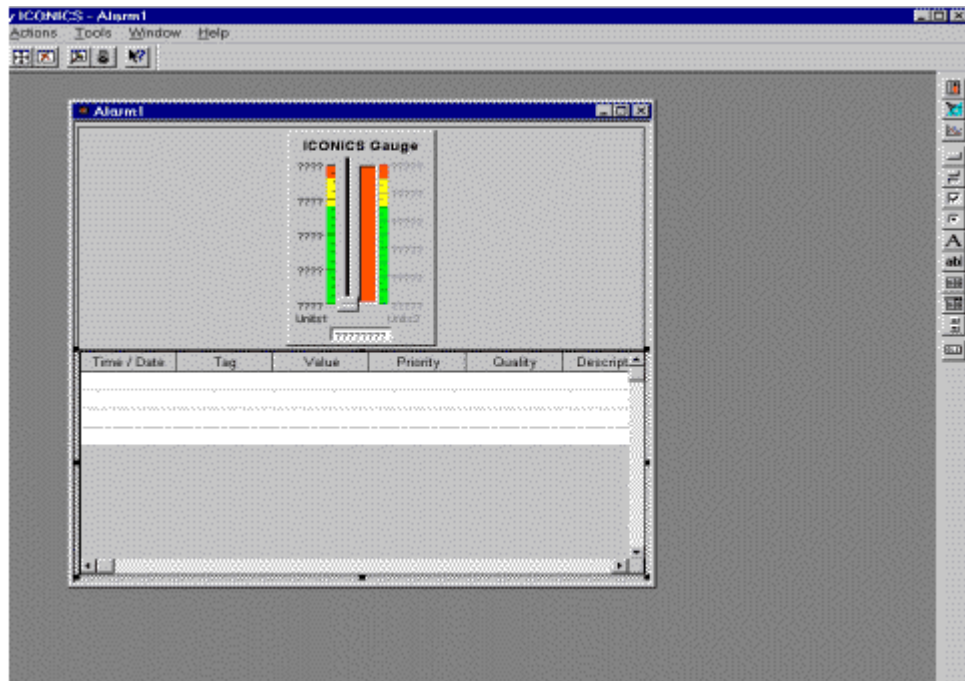
Insert Control

This radio button provides the option to insert an ActiveX control from the list of existing ActiveX object types as shown below:



Insert Object Dialog- Insert Control Tab

Pressing the **Add Control** button enables the ActiveX Control allowing the user to activate the object.



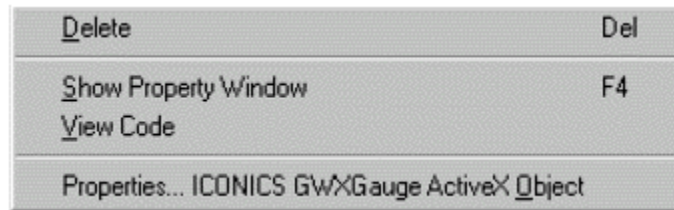
Activated ICONICS Gauge

Links

Selecting this item under the Edit Menu allows the user to establish links between the inserted object and any other file or object.

ActiveX Object Pop-up Menu

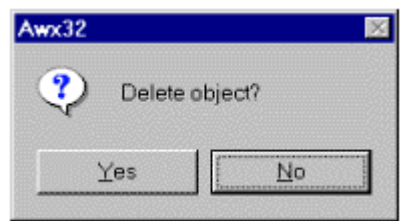
Once the object of choice is inserted in the container, right clicking on the inserted object displays the following menu:



ActiveX Object Menu

Delete

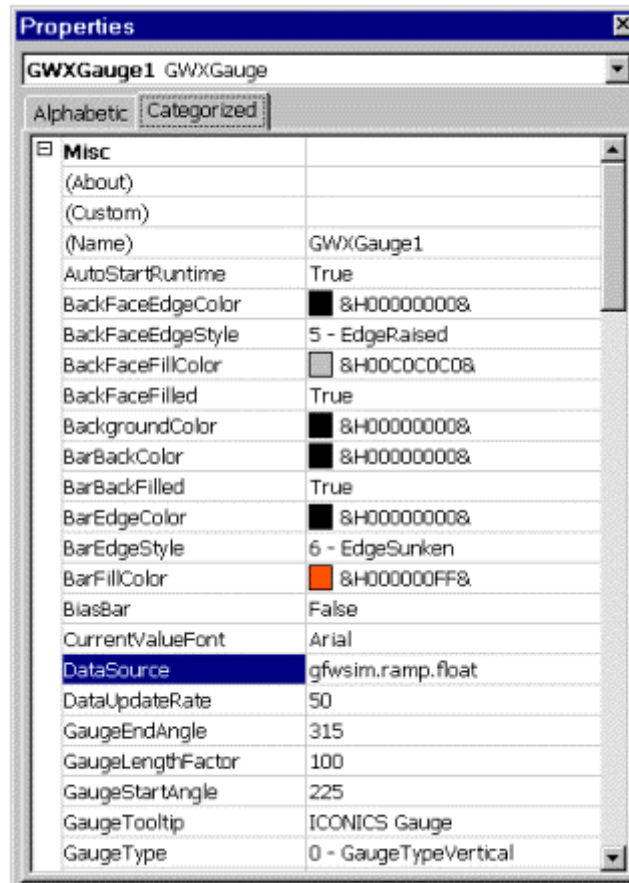
Selecting this menu item simply allows the user to delete the object inserted within the container after the cautionary warning message has been displayed:



Warning

Show Property Window

Selecting the Edit/Show Property Window item displays the Property Window relevant to the object as shown in below:



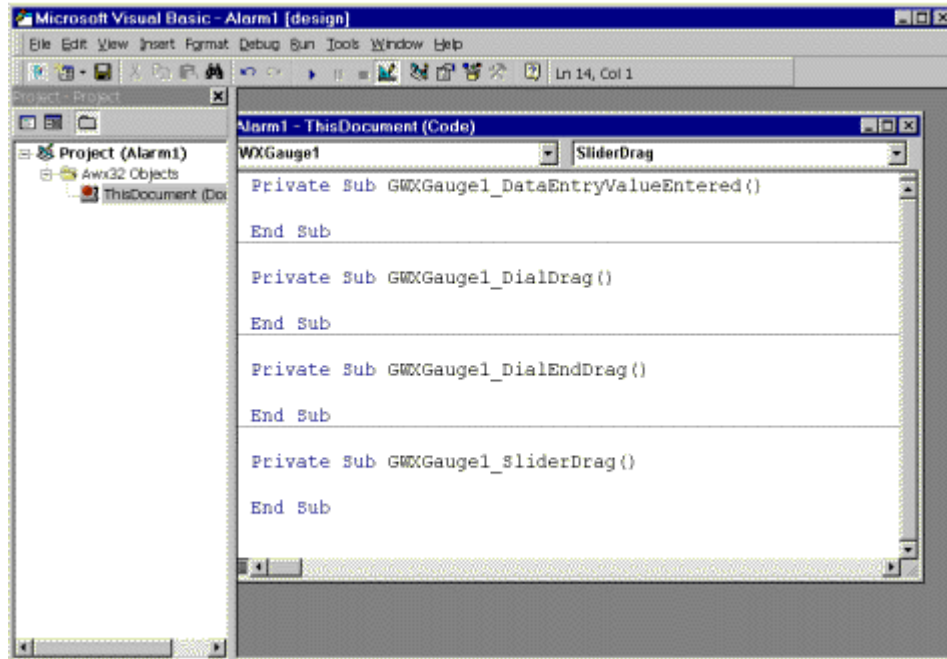
Property Window

This window displays in alphabetic or categorized form, all the properties applicable to the objects within the container. The properties window contains all configurable options for either the selected windows, whether it is the ActiveX object or the ActiveX Control or the entire display.

The user can switch from window to window or from window to display by either: clicking on the desired area that displays the relevant properties window; or by choosing it from the drop down menu at the top of the properties window. To edit an option in the properties window click on that field and either use the available dropdown menu or simply type in your changes. These properties can be also be modified using another user interface - the Properties dialog displayed by selecting the ActiveX Object/Properties menu item under the Edit menu. This item is also accessible through the Edit menu, which comes up if you right-click on the appropriate window.

View Code

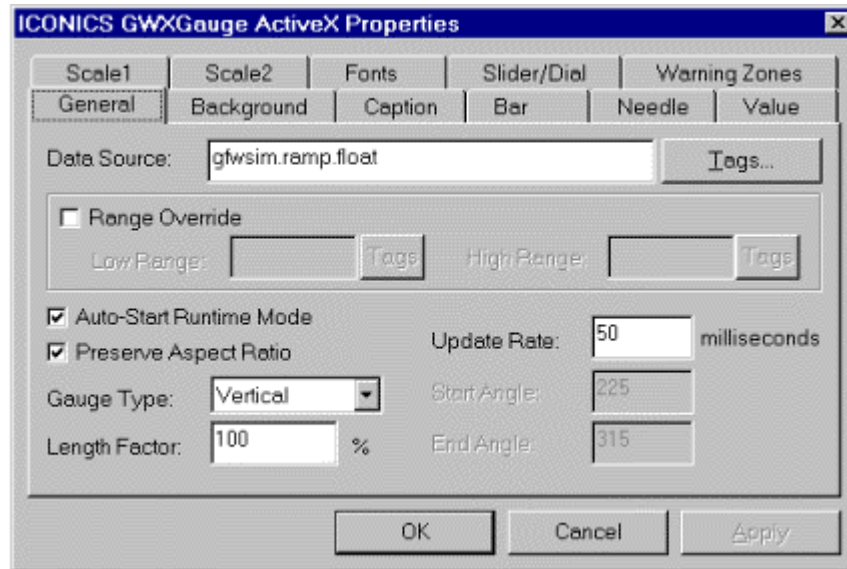
Selecting this item brings up the Microsoft Visual Basic Editor as shown below and allows you to both view and edit the code pertaining to the object.



Microsoft Visual Basic Editor

ActiveX Object - Properties Dialog

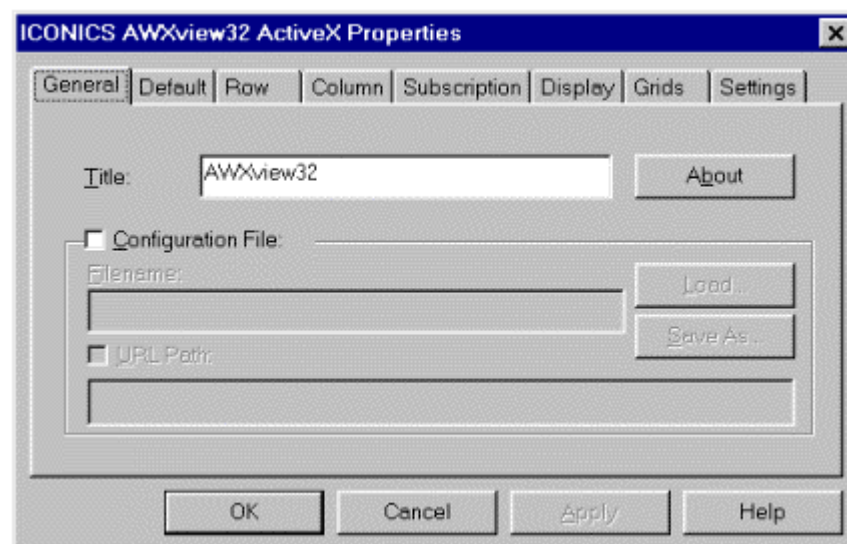
This menu item (specifically displaying the name of the ActiveX object inserted) becomes available in the Edit Menu once an object is inserted in the Container. Selecting the Object/Properties item displays the Properties Dialog relevant to that particular Active X Object. Note that certain ActiveX properties can only be configured from the ActiveX properties Dialog. As mentioned earlier it is also possible to configure the object using the Property Window available through the Edit- Show Property Window menu item:



ActiveX Properties Dialog

This ActiveX Properties Dialog varies greatly depending on the type of ActiveX object inserted into the container. Generally speaking, the tabs in this dialog provide options to edit and alter the display and modify the object. The Tabs generally available in the Properties Dialog for most ActiveX Objects are discussed below. As mentioned earlier this dialog can be displayed by selecting the Edit/Properties Window or by right clicking on the object and using the pop-up menu that displays.

Double-clicking on the AlarmWorX32 Viewer ActiveX control that has been inserted in the AlarmWorX32 container brings up the AlarmworX32 ActiveX Viewer. The Viewer Property Pages are shown in the following figure. This feature is discussed in detail in the AlarmworX32 Viewer chapter:



AlarmWorX32 Viewer

The AWXView32ActiveX Object Properties dialog box contains the following categories:

- General
- Default
- Row
- Column
- Subscription
- Display
- Grids
- Settings

To get a thorough description of each of these sections see the AlarmWorX32 Viewer section.

AlarmWorX32 Container: Other Screen Features

- Introduction**
- View Menu**
- Actions Menu Items**
- Tools Menu**
- Security**
- Window Menu**
- Help Menu**

Introduction

This section discusses the other features on the Menu bar and the Toolbar in the AlarmWorX32 container application. These features are available to the user through the View Menu, the Action Menu, the Tool Menu, Windows Menu and the accompanying icons on the Toolbar. These menus and icons can be used among other things, to alter properties of the object in the container, to exercise display preferences and to modify the appearance of the application. Also accessible through these Screen features are further options to toggle between Runtime and Configure Mode and to customize your system using Macros.

View Menu

The View menu contains the following functions applicable to the appearance of the Object, the Window in your application and your Display:

- Properties Window
- Display Preferences
- Application Preferences
- Main Toolbar
- ActiveX Toolbar
- Status Bar
- Object Layout
 - * Size to Display
 - * Tile to Display
 - * Cascade to Display
- Grid
- Select Language

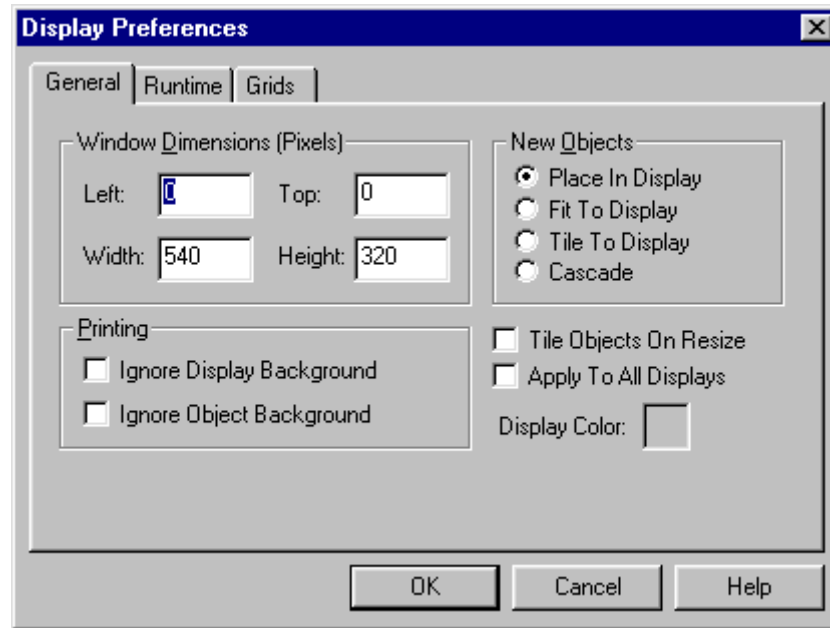
Properties Window
Display Preferences
Application Preferences
Object Layout
Select Language

Properties Window

Selecting this item under the View Menu displays the Properties Window for the object in the container. This feature is discussed in the **ActiveX Object Pop-up Menu** section.

Display Preferences

Selecting this item under the View Menu brings up the Display Preferences Dialog as shown below. This dialog allows you to configure or change the appearance of the display:

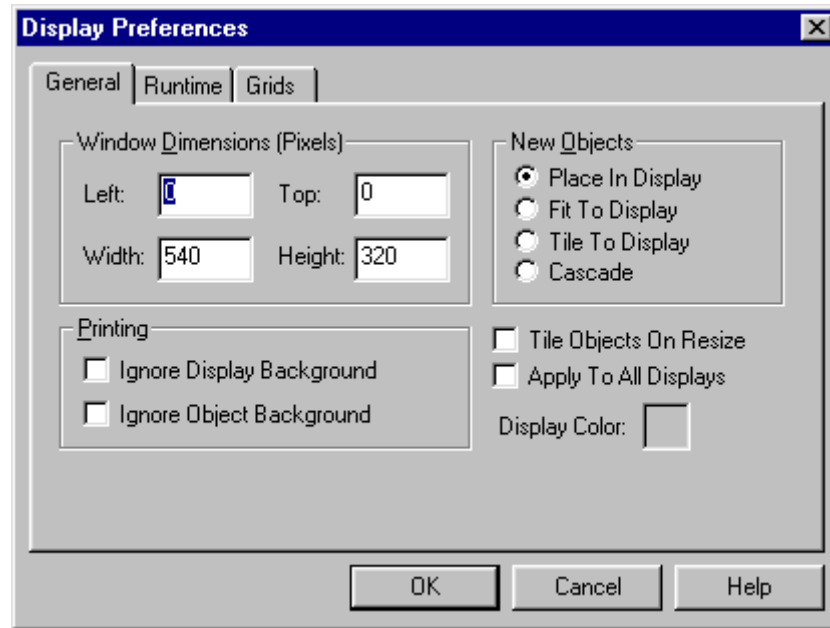


Display Preferences Dialog

General Tab
Runtime Tab
Grids Tab

General Tab

The General Tab allows the user to set the size of his/her display window, choose how to arrange open windows, change the display color and configure the display for printing. This last option has been implemented to save ink while printing displays.



Display Preferences/ General Tab

Window Dimensions

The Window Dimensions field allows the user to specify window dimensions from the left and the top of the window, as well as the width and the height of the window in pixels.

New Objects

The New Objects section allows the user to position the object with respect to the display. It is possible to place the object in the display, size the object to the display, tile the object to the display or arrange the object in the cascade mode.

Printing

Under the Printing section, the user can choose to ignore the display background and/or ignore the object background while printing.

Apply to All Displays

If checked, the parameters defined in the Display Preferences/ General Tab dialog will be applied to all displays.

Tile Objects on Resize

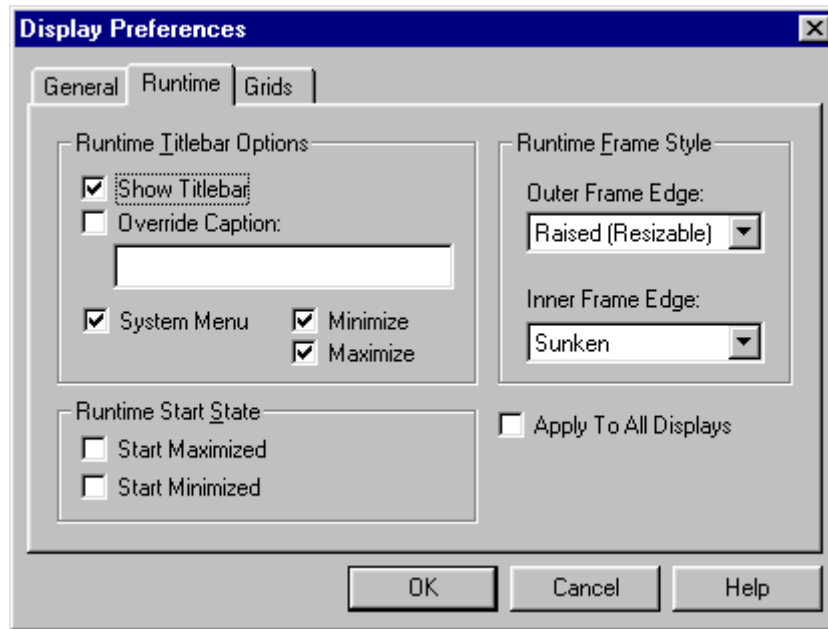
If checked, the Tile objects on Resize option, automatically tiles objects when the display is resized.

Display Color

The Display Color field allows the user to choose a color for the display.

Runtime Tab

The Runtime tab allows the user to configure how the window will appear in runtime.



Display Preferences/ Runtime Tab

Runtime Titlebar Options Section

If the 'Show Titlebar' box is checked the title bar will be displayed in runtime. Similarly if the 'Override Caption' box is checked the user may enter a caption to appear in runtime. The 'System menu', 'Minimize', and 'Maximize' boxes all refer to buttons that may appear in the title bar during runtime. The last two of these options are helpful in ensuring the functionality of the window during runtime because if checked they do not allow the operator to close or change the size of the window.

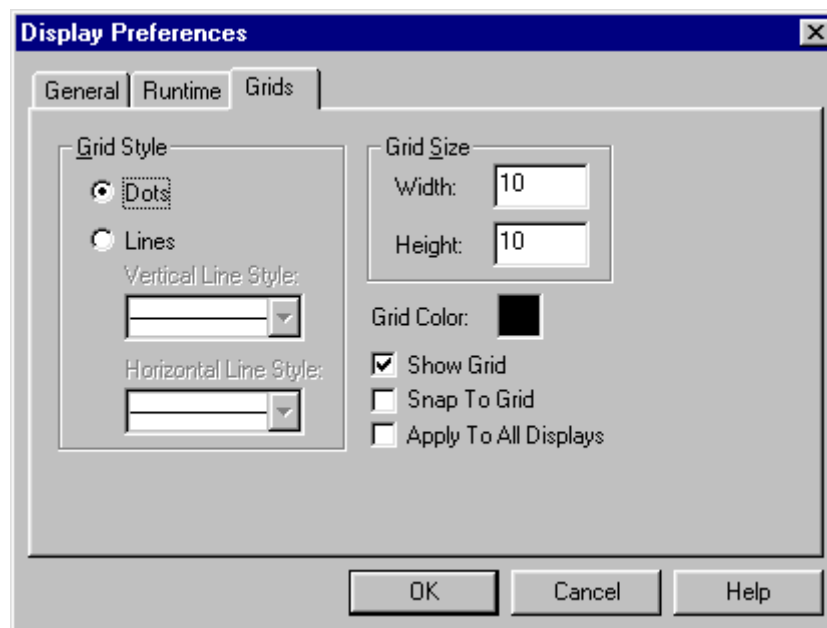
Runtime Frame Style Section

The Outer Frame Edge field allows the user to choose from the following three frame styles: None, Raised (Resizable) or Raised (Fixed Size). Having the option of fixed size is helpful in maintaining the functionality of the trend window in runtime, by not allowing the operator to change the configured display.

Runtime Start State Section

The runtime start state section allows the user to determine how the trend window will start out in runtime, either: minimized or maximized.

Grids Tab



Display Preferences/Grids Tab

The grids tab allows the user to configure the type, style, size and color of the display grid. The 'Show Grid' box makes the grid visible and the 'Snap To Grid' button makes all borders of the display objects align with the grid lines.

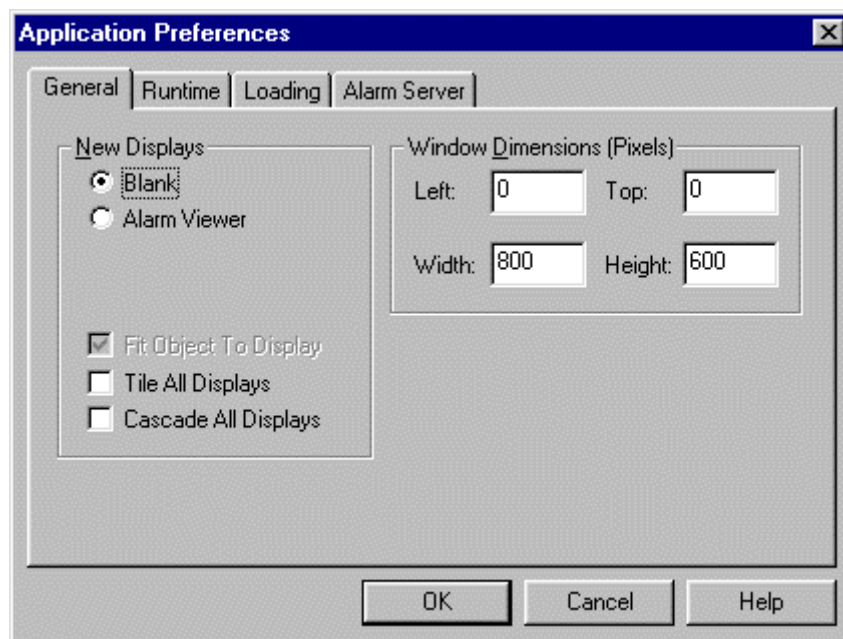
Application Preferences

Choosing to view the Application Preferences for the AlarmWorX32 Container presents the user with four tabs: General, Runtime, Loading and Alarm Server. The settings made on these four tabs will determine how the Container will look when started, how it will start and how it will behave during runtime.

To configure or change the application preferences select 'Application Preferences' from the View Menu.

General Tab
Runtime Tab
Loading Tab
Alarm Server Tab

General Tab



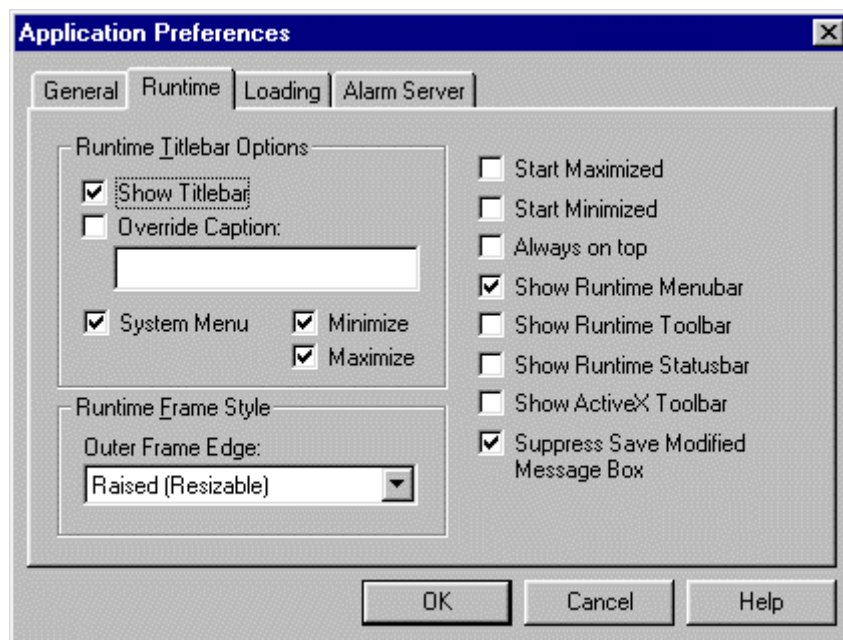
Application Preferences/ General Tab

The Application Preferences **General tab** allows the user to configure new displays. He can choose to have the new display appear either as a blank window or he can opt for a display showing the Alarm Viewer.

The user can define how a new object will be placed in the display using the **Fit Object to Display**, **Tile Display** or **Cascade all Displays** options.

Additionally, the user is able to define the size of the window using the **Window Dimensions** field.

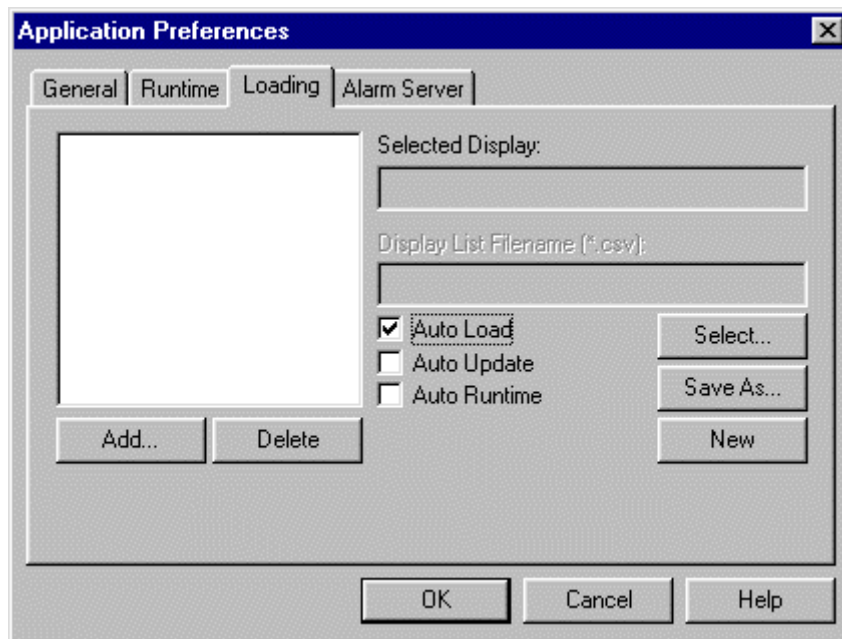
Runtime Tab



Application Preferences/ Runtime Tab

The Application Preferences Runtime tab is similar to the Display Preferences Runtime tab with the exception of the right side options. These options control how the display will look when put into runtime including starting minimized/maximized, always on top, show runtime menu bar, show runtime toolbar, show runtime status bar, show ActiveX toolbar and suppress save modified message box. It is also possible to choose how the titlebar will appear in runtime as well as selecting the runtime frame style.

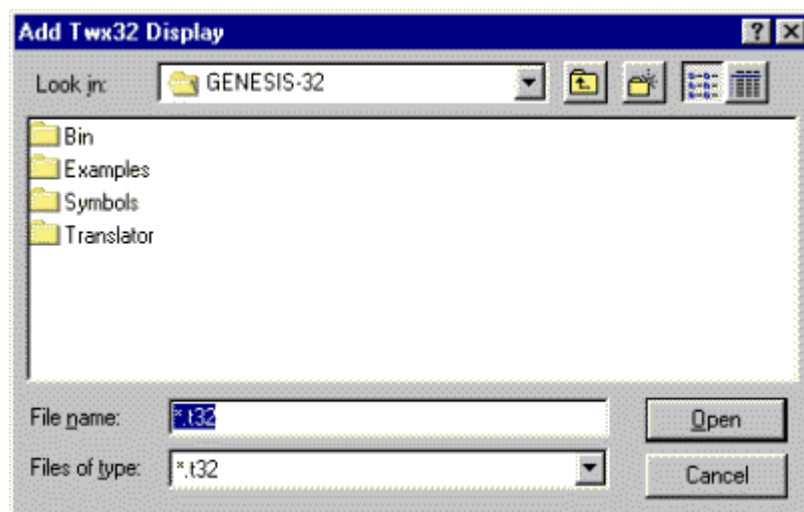
Loading Tab



Application Preferences/ Loading Tab

The loading tab allows the user to add or delete previously created displays to/from his/her window. It is possible to choose to automatically load a certain file or display using the 'Auto Load ...' checkboxes on this tab.

Clicking the **Add** Button brings up the following dialog shown below:

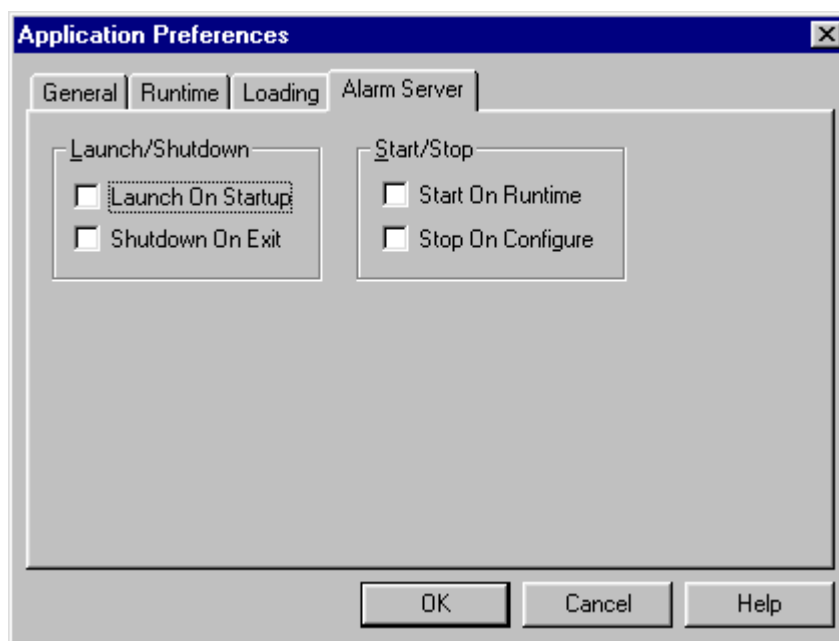


Add Display Dialog

The user can browse the directory for the display he chooses to add. The name of the **Selected display** and the **List file name** with a *.csv extension, displays in the boxes at the right. By checking the **Autoload**, **Autoupdate**

and **Auto Runtime** check boxes, the user can exercise the options to automatically have the new display come up with the added display, to automatically have it appear in its updated/configured mode, and for runtime to commence automatically on opening the new display.

Alarm Server Tab



Application Preferences/ Alarm ServerTab

The alarm server tab allows the user to configure the behavior of the Iconics alarm and events server. The server will behave according to the following selections:

Launch on Startup – Launches the alarm and events server on start of the Alarm container application.

Shutdown on Exit – Will attempt to shutdown the alarm and event server on the shutdown of the container. Other applications may still keep the server running.

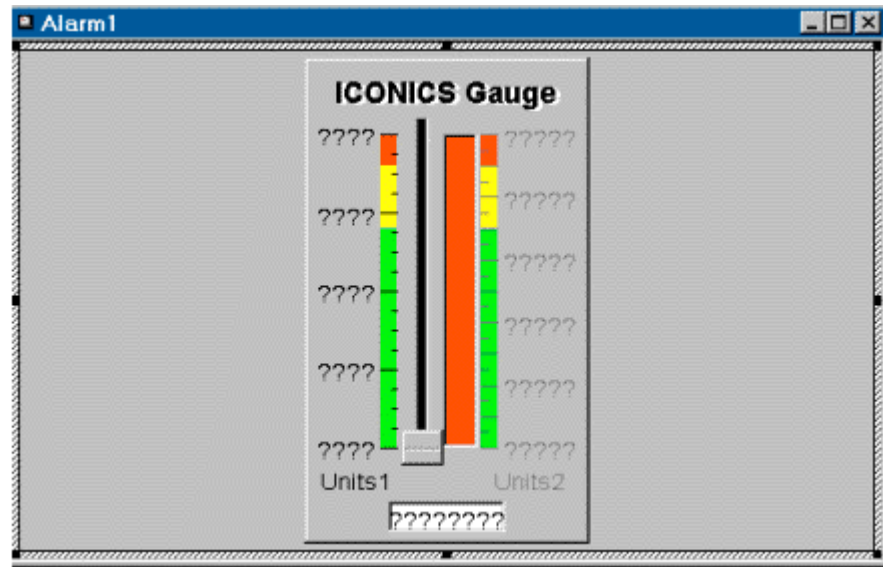
Start on Runtime – Starts the alarm and events server when the alarm container enters runtime.

Stop on Configure – Will attempt to shutdown the alarm and event server when the alarm container exits runtime. Other applications may still keep the server running.

Object Layout

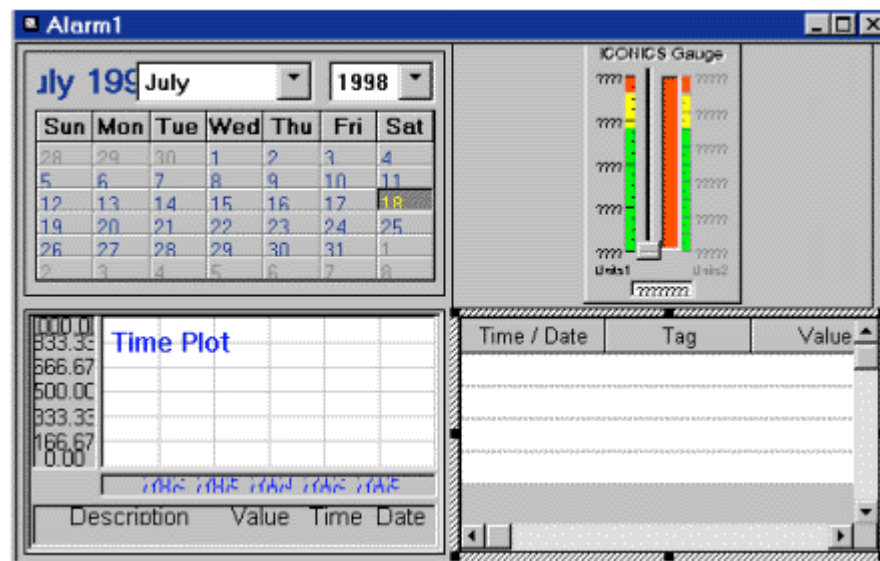
To define how current objects in your display are shown select 'Object Layout' from the view menu and select one of the following: Size to Display, Tile to Display or Cascade.

The **Size to Display** fits the currently selected object to the outer edge of the display window. This item is not available if you do not have an object selected.



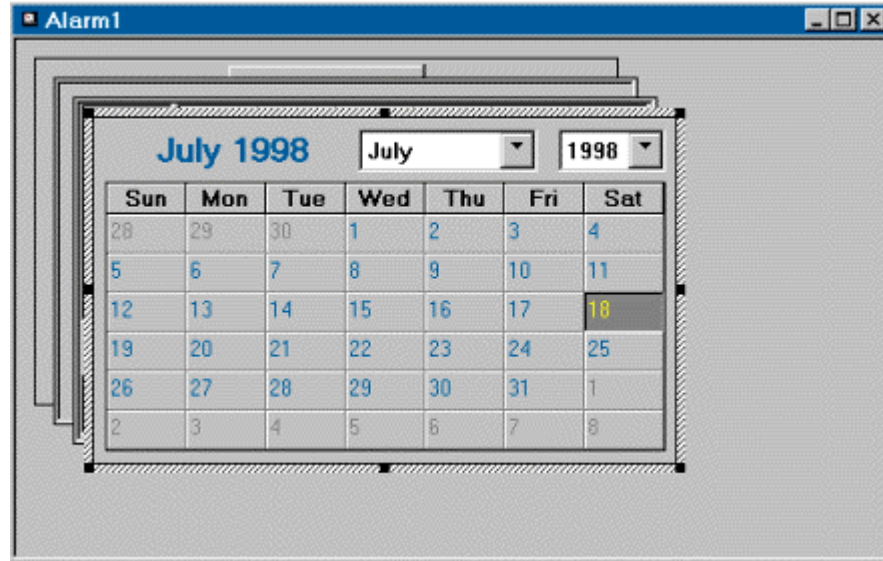
Object Sized to Display

Title to Display changes the size of your objects to fit them into the evenly into the display window. This is helpful if you want to compare data in different objects or if you are moving back in forth from object to object.



Object Tiled to Display

Cascade simply resizes all of the open objects and piles them on top of each other so that you can see the top and left side of every open object window.



Objects in Cascade Mode

Select Language

Selecting the 'Select Language' from the view menu opens the standard language selection dialog box shown below.



Select Language Dialog

The 'List' section allows the user to narrow down the list of languages provided in the list box on the left-hand side of the dialog. The user is able to select one of the top three options: English, Localized or Native. Additionally, the user is able to select to view only Installed Locales, Only Available Language Translations or both. If neither of these options is selected then the user will get a list of languages that may or may not be available on their machine.

Actions Menu Items

The Actions menu allows the user to apply two different actions to their display.

- Animate Mode - Active Display
- Runtime Mode – Entire Application

1. Animate Mode

Selecting the Animate Mode menu item will toggle the active display in and out of animate mode.

2. Runtime Mode

Selecting the Runtime Mode menu item allows the user to put the application into runtime or configure mode.

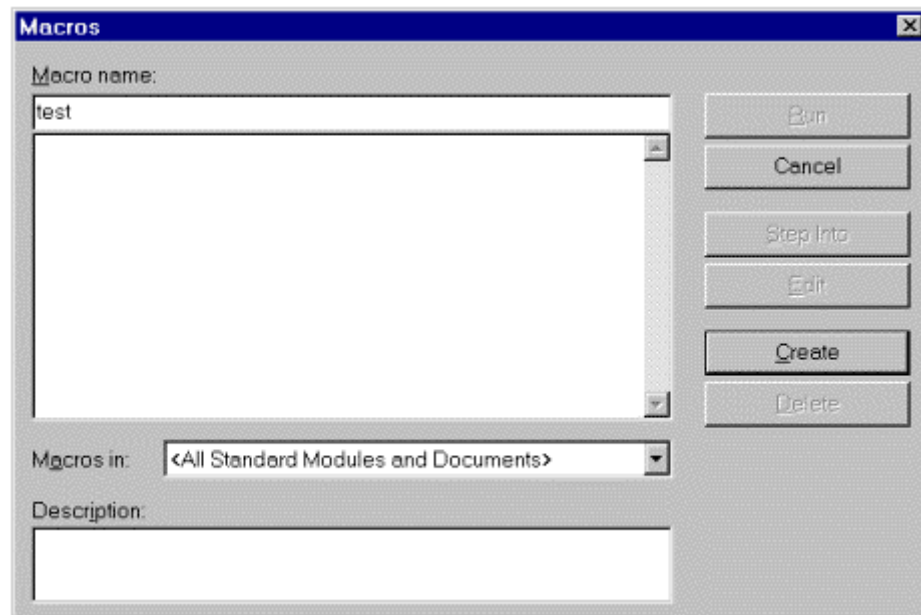
Tools Menu

The Tools menu contains the following functions, which are useful in customizing your system:

- Macro
 - * Macros - Alt + F8
 - * Record New Macro
 - * Visual Basic Editor
- Launch Configurator – F9
- Logger Configurator
- Security
 - * Login/ Configuration - F7
- Set Working Directory - F8

Macro

Selecting this item under the Tools Menu, allows you to run a Macro using the standard interface shown below:



Macros Dialog

Pressing the **Create** button brings up the Microsoft Visual Basic Editor where you can create a new macro. You can an existing macro by entering the name of the macro in the box: **Macro Name** and clicking the **Run** button.

Record New Macro

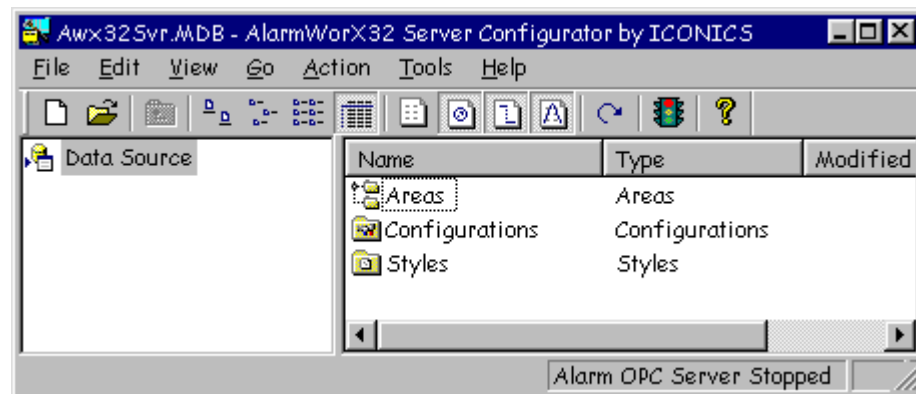
Selecting this item under the Tools Menu displays the Visual Basic Editor where you can record a new macro.

Visual Basic Editor

Selecting the **Visual Basic Editor** item under the Tools Menu displays the same.

Launch Configurator

Selecting the Launch Configurator menu item from the Edit Menu brings up the Iconics Alarm Server Configuration Toolas shown in the figure below:



Launch Configurator

The Configurator has been discussed in detail in the Configurator chapter of this manual.

Logger Configurator

The logger configurator menu item brings up the Iconics Logger configuration tool.

Security

The ICONICS Genesis32 Security System provides restricted access to Genesis32 functions based on the concept of a logged in user. A Security System Administrator configures the system by adding users and assigning them specific Genesis32 privileges. In addition, Administrators may associate users with certain Administrator defined groups that also have assigned privileges. Thus a user has the effective rights of all the groups to which he belongs plus his own private rights.

The user/group concept for security assignment is well established in computer operating systems (such as Windows NT) and computer networks (such as Novell Netware). It is assumed that the reader has an understanding of these concepts.

The Security item under the Tools Menu allows the user to establish the security configuration for the AlarmWorX32 application.

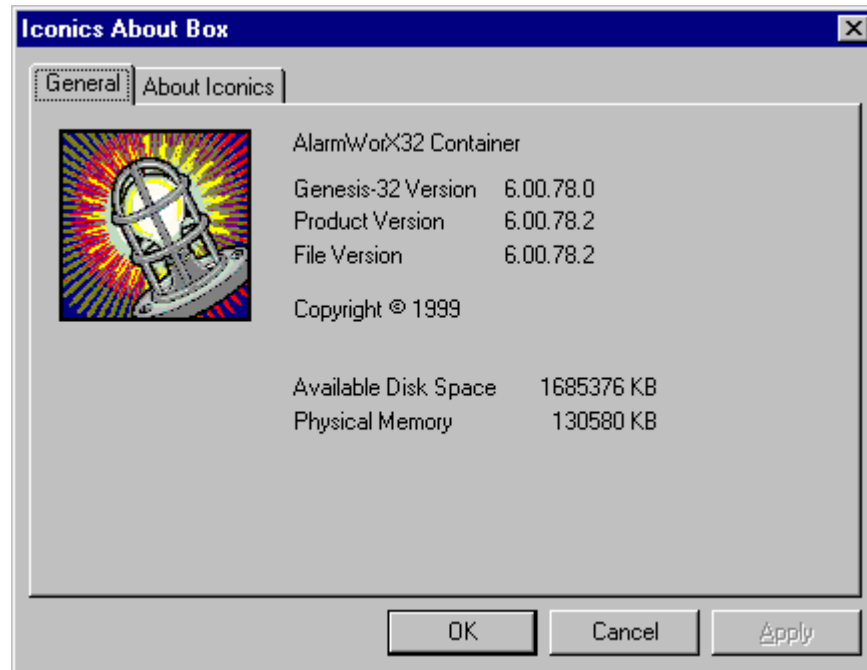
For a detailed discussion on how to configure the Security Server and for further reference on Security for Genesis32 applications, consult the Security Help.

Window Menu

The Window menu displays a drop-down list of all open windows. The window currently displayed is preceded in this list by a check mark. You can display a window by selecting it from this list. You can also opt to close all windows. Also available under the same menu are items that allow the user to modify the appearance of the display, to arrange windows in **Cascade** or **Tile** format.

Help Menu

The Help menu contains an online tutorial that you can use to become familiar with the most common features of AlarmWorX32. The Help menu also displays both technical support information and the AlarmWorX32 About box. The About box contains registration and serial number information, plus system resource information, such as memory and disk space available.



Iconics About Box



Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.

Foxborough, MA 02035

Tel: 508-543-8600

FAX: 508-543-1503

E-MAIL: support@iconics.com

WEB: <http://www.iconics.com>

Contents

Introduction

Using AlarmWorX32 Viewer ActiveX

Connections

Security

Language-Aliasing Support

Asynchronous Downloading Features

User Interface

Runtime Operations

Introduction

The AlarmWorX32 Current Events Viewer is an OPC ActiveX Client application that provides real-time alarm information within the Genesis32 product family. It communicates to OPC Alarm and Events Servers to provide excellent alarming features. It is important to note that this is a **Current Events** Viewer, which means that it only displays active alarms and current operator messages.

The structure of the AlarmWorX32 Current Events Viewer is built on a three level system, consisting of Default settings (color, font, justification, wrapping method and size), Row types (color and font) and Column values (color, font, justification, wrapping method and size). Each system starts with certain default settings, which can be over written by row settings, which in turn can be over written by column settings.

Using AlarmWorX32 Viewer ActiveX

The AlarmWorX32 Viewer ActiveX integrates easily within Visual C++ or Visual Basic container applications. The ICONICS Genesis32 containers, such as GraphWorX32, provide Toolbar support for inserting the AlarmWorX32 Viewer ActiveX. Once inserted into a document or a form the component can be resized and configured.

Within the ICONICS Genesis32 containers and in Visual Basic forms, the component is initially placed in CONFIGURE mode, where its user interface is disabled and the user can configure the AlarmWorX32 Viewer ActiveX through the property pages. Once the application is committed to Runtime the viewer will automatically connect to alarm and event OPC servers, query for a complete update of all requested alarms and display current alarm messages. The AlarmWorX32 Viewer ActiveX has the option of saving file information as part of the container or in its own file format.

The user must give a valid filename, if saving to a file, before the data are serialized and saved. The AlarmWorX32 Viewer uses the concept of a "Working Directory", defined in the ICONICS Genesis32 containers. This represents the working directory where project files and configuration files will be stored. AlarmWorX32 Viewer will utilize this directory as a starting point for saving and loading its configuration files.

Connections

The AlarmWorX32 Viewer can display data from any OPC Alarm and Event server. Alarm subscription filters are made through the subscription property page.

Once a subscription is set up the viewer requests a refresh of all current alarms upon animation. Alarms are sent by the OPC Alarm and Event Server, on an alarm by alarm basis. This lowers the number of alarms sent to each viewer and allows for server side filtering. Examples of subscriptions can be found in section 6.7.1.5 of the user manual.

Security

To access the ICONICS security server either select **Tools - Security - Configuration** or click the Security icon from the toolbar and the following dialog will appear.

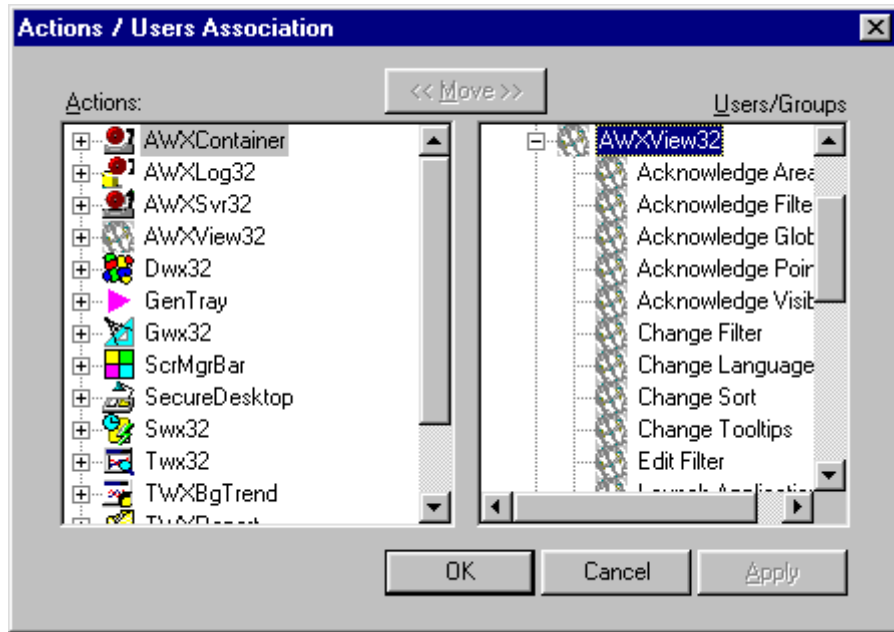


The dialog box is titled "ICONICS Security Server Administrator Login". It contains a lock icon and the following text: "Enter an Administrator user name and password, or leave the user name blank and enter the default Administrator password:". Below this text are three input fields: "User Name:", "Password:", and "Challenge:". The "Challenge:" field contains the value "18303". At the bottom are "OK" and "Cancel" buttons.

Enter your User Name and Password and click OK. This will bring you to the ICONICS Security Server where it will be possible to access the AWXview32 Security Server. Depending on your configured level of security you may or may not be able to edit components of the AWXview32 Security Server.

Note: For more information about security please see the Security chapter of the AlarmWorX32 manual.

Once a user has gained access the ICONICS Security Server dialog box will appear. Under Edit choose Application Actions and the Actions/Users Association dialog box will appear.



The security server can be set up to allow access to several different actions from this dialog. To set the options for the AlarmWorX32 Viewer select AWXView32 from the list in the right pane. This will bring up the list seen in the left pane, which includes the following options:

- Acknowledge Area
- Acknowledge Filtered
- Acknowledge Global
- Acknowledge Point
- Acknowledge Visible
- Change Filter
- Change Language
- Change Sort
- Change ToolTips
- Configure Property Pages
- Launch Application
- Launch Displays
- Show - Hide Columns
- Trigger VB Events
- View Details

Language-Aliasing Support

The AlarmWorX32 Viewer ActiveX component has all of its strings saved in the resource file, which can be modified to provide custom versions. In addition, it provides real-time support for loading a resource-only .dll to support an international language. This .dll file is compiled as a standard Win32 .dll, and it is made up of all the resources and the .odl file of the AlarmWorX32 Viewer module. Any time the ICONICS applications settings are switched to a different language, AlarmWorX32 Viewer will load the corresponding resource .dll.

The resource-only .dll file name uses the AlarmWorX32 Viewer component, followed by the three characters used in Windows NT as the abbreviation for a language name (e.g., the abbreviation "deu" stands for German, and the .dll should be named "AWXView32_deu.dll"). In addition, using the OLE Automation property language ID, you can control this functionality through OLE Automation.

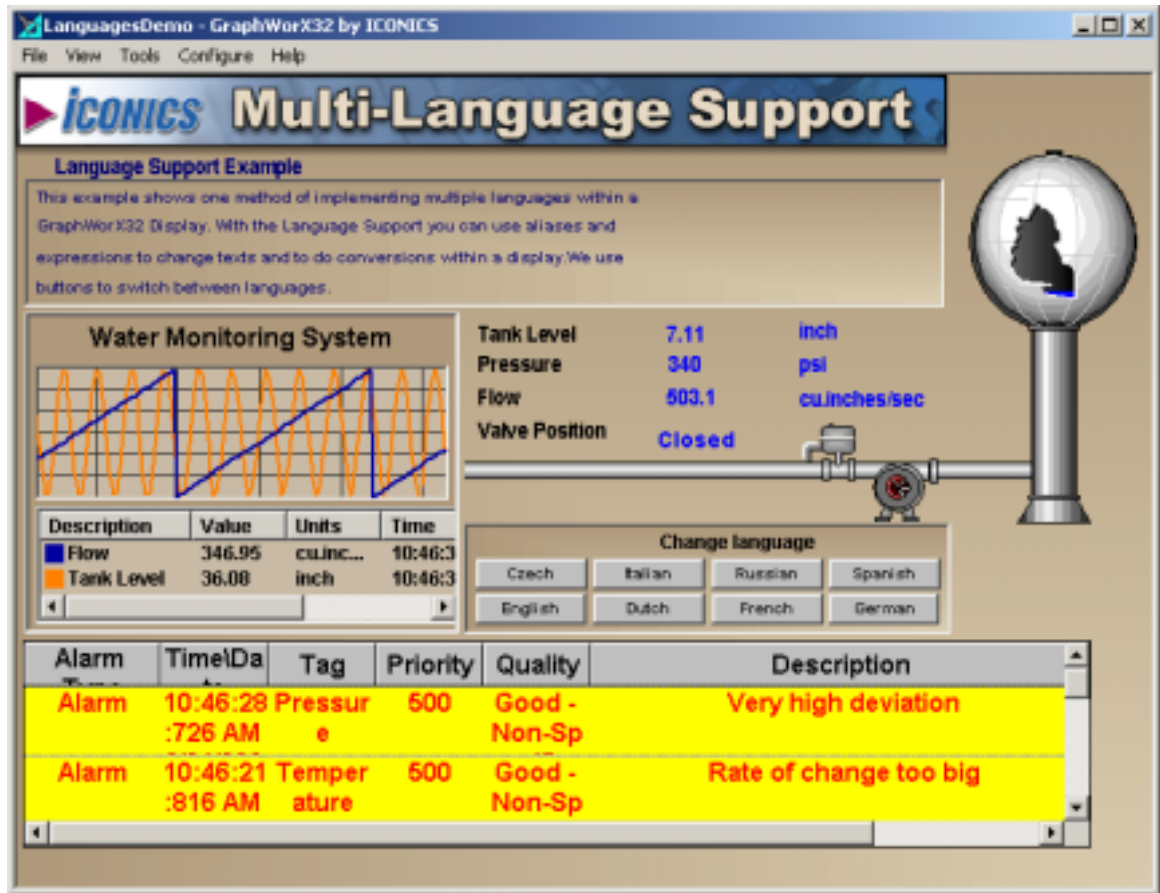
The newly added Unicode version of the AlarmWorX32 Viewer ActiveX also supports language aliasing with automatic range and value scaling, as well as unit conversions, formatting, and font selection. For more information about configuring Unicode version language-aliasing support, refer to the GENESIS32 **Language Configurator** documentation.

Note: The AlarmWorX32 Viewer requires the following additional language-aliasing configuration work. If any expression is to be used for range, alarm limits, and or unit scaling, you need to define both a "Read Expression" and a "Write Expression" in the Language Configurator. By doing so, the Viewer will be able to function properly between language switches.

Note: To enhance language-aliasing performance, select the Microsoft Arial Unicode font, which contains all Unicode characters. The Arial Unicode MS font must be selected separately within the Properties dialog boxes of each ActiveX component.

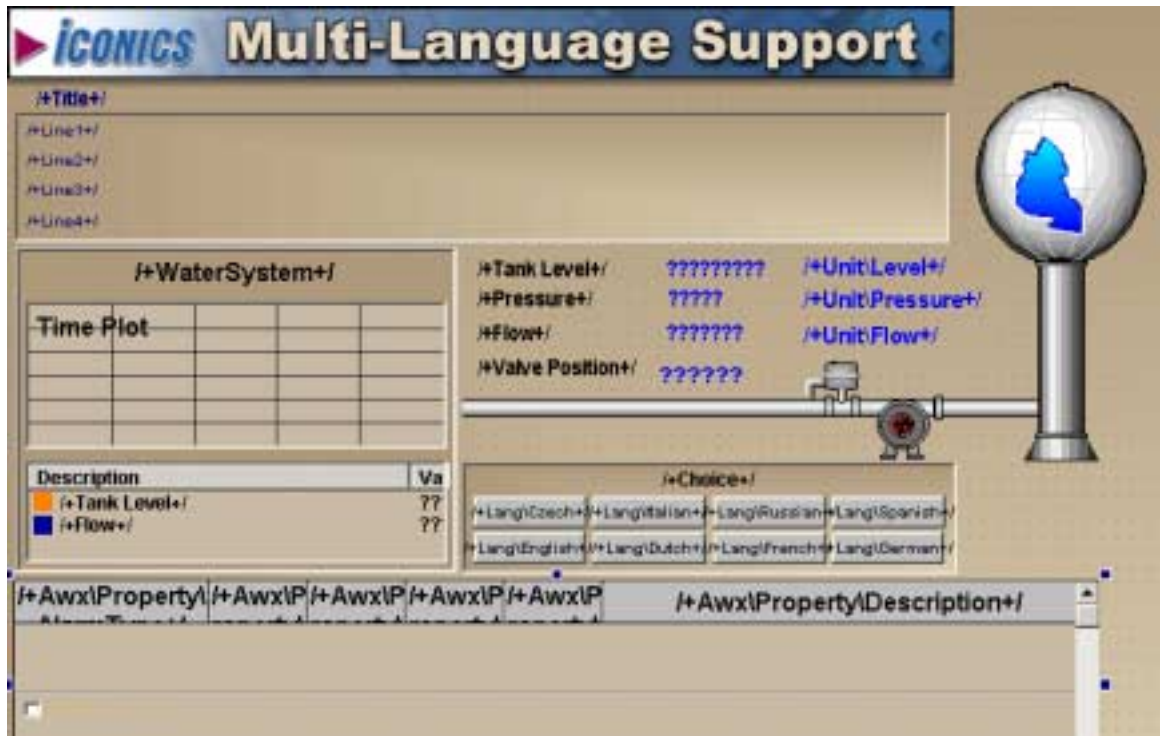
Sample Language Configuration

The Unicode Installation of GENESIS32 Version 6.10 installs a sample Language Server configuration under the "Languages" folder of the GENESIS32 product installation "tree." In addition, a sample display "languagesDemo.gdf," which includes a sample Viewer configuration, is provided under the GEN32DEMO folder. The figure below shows this sample during runtime mode. Notice that you can switch between languages in this display by simply clicking the buttons in the Change Language field.



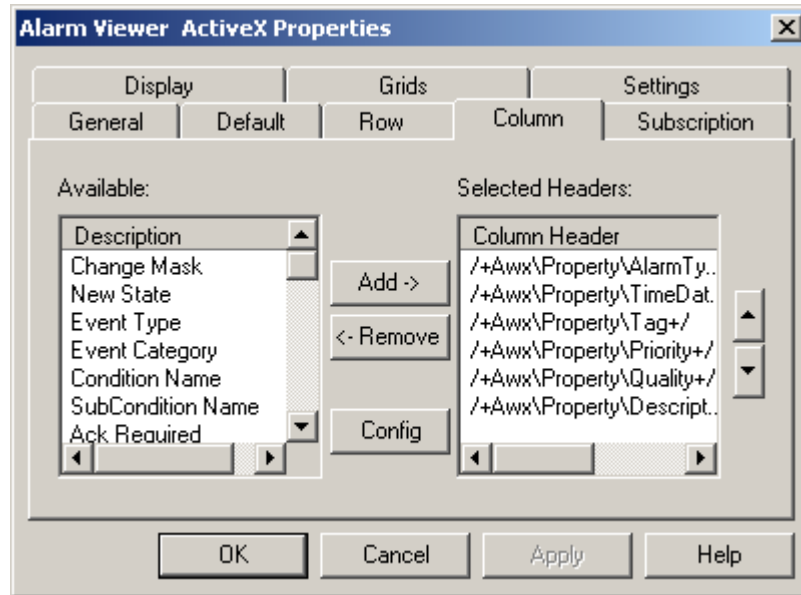
Language-Aliasing Demo in Runtime Mode

The figure below shows this same language-aliasing demo in configuration mode. This display incorporates ActiveX objects from GraphWorX32, TrendWorX32, and AlarmWorX32.



Language-Aliasing Demo in Configuration Mode

For example, if you click on the Alarm Viewer ActiveX in the language-aliasing configuration shown above, you can view the properties of the Alarm Viewer Active X, as shown in the figure below. The **Column Tab** of the **Alarm Viewer ActiveX Properties** dialog box, shown below, displays the language-aliasing configuration of the headers in the Alarm Viewer.



Alarm Viewer ActiveX Properties: Column Tab

For the example above, the Alarm Viewer configuration appears as shown in the figure below. Note that the strings within a "/" and "+" delimiter pair define a language-aliased string. As you can see, each header in the Alarm Viewer configuration is a language-aliased string that contains these delimiters. During animation (runtime) mode, the Viewer ActiveX will interface to the ICONICS Language Server, and it will try to resolve the language-aliased strings. If successful, it will display the translated strings in the Alarm Viewer display. In addition, if the **Units** language alias string has an equation defined in the Language Configurator, it will autoscale the values based on the scaling equation supplied by the Language Server.

/+Awx\Property\	/+Awx\Property\TimeDat	/+Awx\Property\Tag	/+Awx\IP	/+Awx\Prope
Alarm Type	Time	Tag	IP	Property

Language-Aliased Alarm Viewer Display in Configuration Mode

During animation (runtime) mode, the Viewer ActiveX will interface to the ICONICS Language Server, and it will try to resolve the language-aliased strings. If successful, it will display the translated strings in the Alarm Viewer display, as shown below. In addition, if the **Units** language alias string has

an equation defined in the Language Configurator, it will autoscale the values based on the scaling equation supplied by the Language Server.

AwxlProperty\AlarmType	AwxlProperty\TimeDate	AwxlProperty\Tag	AwxlProperty\Value
Alarm	08:59:54:923 19.03.2001	Pressure	500
Alarm	09:00:00:251 19.03.2001	Temperature	500

Language-Aliased Alarm Viewer Display in Runtime Mode

Note: The Unicode version language aliasing is independent of the resource .dll selection, which relates mostly to the text in the dialog boxes and other user interface elements. The Unicode version of GENESIS32 is available only for Windows 2000 and Windows NT.

Asynchronous Downloading features

The AlarmWorX32 Viewer supports asynchronous downloading of configuration data. When used inside ICONICS' ActiveX Document servers, such as GraphWorX32, it can be viewed using Internet Explorer.

To enable this feature, check the Use URL Path checkbox and supply a URL path, on the General property page. In case of viewing ActiveX documents within an Internet set up, the user can supply a computer name and directory specifying the location of the AlarmWorX32 Viewer configuration data (must be a shared directory), by using the convention:

<file://Computer Name\\Directory\\filename.a32>

User Interface

This section describes the user interface for AlarmWorX32 Viewer ActiveX.

Properties Dialog Client Side Filtering

Properties Dialog

The AlarmWorX32 Viewer property page dialog is the user configuration interface to the ActiveX. When embedded in an Iconics ActiveX container, double clicking anywhere in the ActiveX client during configuration causes the property page dialog to be displayed.

Note: By right clicking on the Viewer and selecting 'Show Property Page' the Property Page will appear.

General
Default
Row
Column
Display
Subscription

General

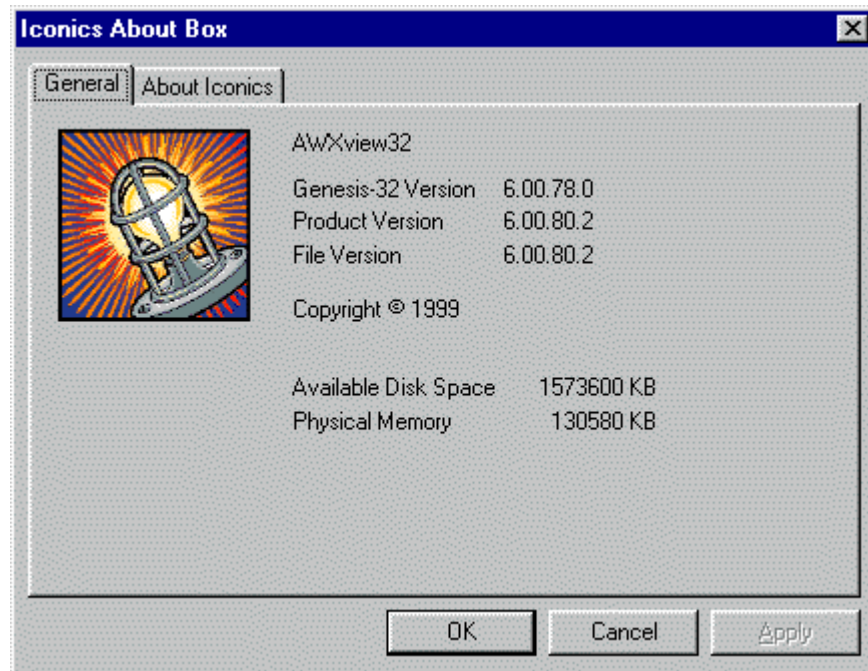
The screenshot shows the 'General' tab of a properties dialog. At the top, there is a 'Title' text box containing 'AWXview32' and an 'About' button. Below this, there is a checked checkbox labeled 'Configuration File'. Underneath the checkbox is a 'Filename:' text box and a 'Load...' button. Below the 'Filename' box is an unchecked checkbox labeled 'URL Path:' and a 'Save As...' button. At the bottom, there is an empty text box.

By default information is stored as part of the container application's document file.

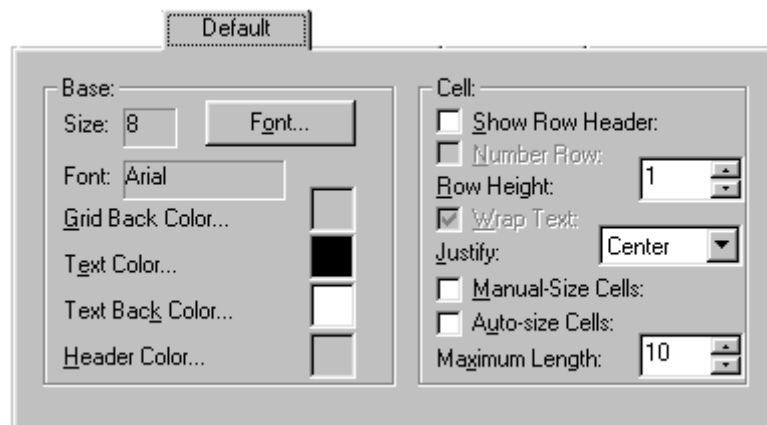
If the Configuration File check box is selected, information is stored as part of a separate file. Select Save and enter the path and file name. The results will be displayed in the 'Filename' field. If 'Configuration File' is selected the viewer will attempt to load from the file listed in 'Filename'. If 'Configuration File' and 'URL Path' are both selected the ActiveX will make use of the URL path on loading.

Note: the ActiveX can not save to a URL path. To create a URL file, save a configuration to a local file and copy the file to the desired network location.

Click the 'About' button to view information on this product, including: Genesis32 Version, Product Version, File Version, copyright information, available disk space and physical memory. Open the 'About Iconics' tab to view contact information for ICONICS.



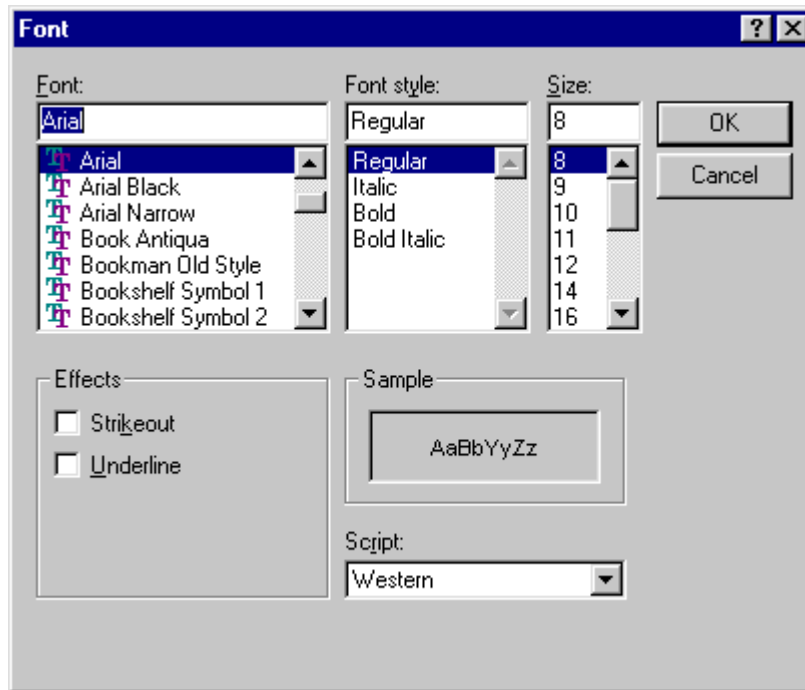
Default



The Default Tab allows the user to configure the appearance of their AlarmWorX32 ActiveX.

Font

The Font section allows the operator to determine the font size, style and effects. To set these features click the 'Font' button and the following dialog will appear.



Cell

The Cell section allows the user to choose which default options they want to apply to the cells of their Viewer. Selecting the 'Show Row Header' box will make the 'Number Row' option available but it is not necessary to have both of these options checked. If the 'Number Row' option is not selected then the row headers will be blank. The 'Row Height' field allows the user to set the height of the rows in their Viewer. 'Show Row Header' causes a row header to be displayed. 'Number row' causes the row header to number each row line.

Note: The 'Row Height' must be a positive integer between 1 and 10.

Checking the 'Wrap Text' box makes the text wrap when it hits the end of the particular cell. When this option is not applied text will continue in one line although only part of the text will be visible, depending on the size of the cell. Please note that the 'Wrap Text' feature will only be visible with multi-line rows. The 'Justify' field will determine the justification (left, center or right) of the text within each cell.

There are three options for determining the size of each cell:

- Manual-Size Cells

Checking this box indicates that the user would like to be able to set the width of each column. Manual sizing of columns can only be done during runtime.

- Auto-Size Cells

Checking this box indicates that the user would like the Viewer to set the width of each column. The AlarmWorX32 Viewer will automatically set each column to the width nearest to the size of the ActiveX divided by the number of visible columns

- Entering a maximum length for each cell.

Currently this is a range from 1 to 1000 and applies the maximum cell length to all cells.

Colors

The Colors Section allows the user to chose colors for the background of the display, the background color of the text, the background color of the Header and the Base foreground color Text. To change any of these colors simply click on the corresponding color box and a Color text box will appear allowing the user to chose a pre-configured color or to create their own color using the 'Define Custom Color' button.

Row

Row

Event Type: Alarm / Condition

Severity: 0 - 1000

Configure

Color:

Text Color... [Yellow Box]

Background Color... [Red Box]

Font:

Size: 8

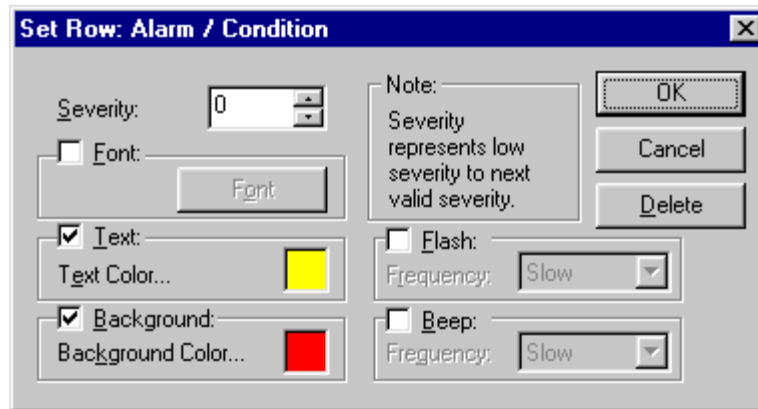
Font: Arial

Actions:

Flash: Disabled

Beep: Disabled

The Row tab allows the user to set the color of the text and background for different event types. Use the Event Type drop down menu to select the event type you wish to configure. It is possible to have several different configurations for one event type. To configure an event type, simply click the 'Configure' button or click either the 'Text Color' or 'Background Color' box and the following dialog will appear.



The 'Severity' field allows the user to set the lower bound for this particular configuration. This lower bound represents the point at which AWXview32 will begin to override the current values for text color, background color and font with the values set by the user in this window.

Note: The new height will be resized based on the size of the font selected.

Through this method it is possible to set up ranges. For example configuring a severity of 300 with a text color of purple and another severity of 600 with a text color of green would result in a purple range from 300 to 599.

This dialog can also be used to delete a configuration. To delete a configuration, select the lower bound severity of the configuration you wish to delete and click the 'Delete' button.

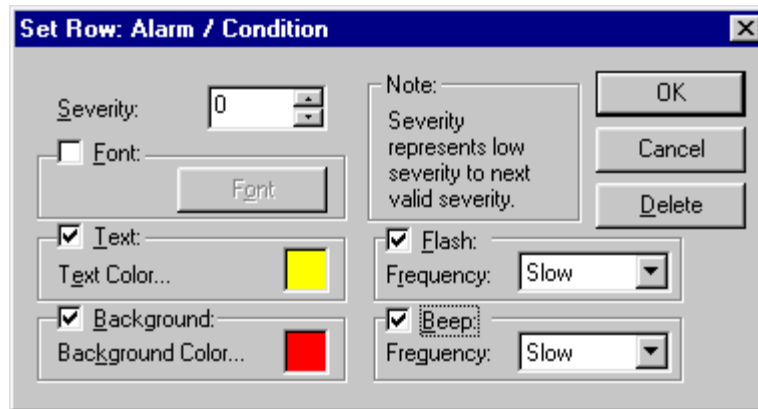
Note: The default severity value '0' can not be deleted.

Selecting the 'Flash' and/or 'Beep' boxes will apply the flash or beep to the particular severity. Once the box has been checked, the user is able to set a frequency for the occurrence of the flash or beep (slow, medium or fast). This frequency will be displayed in the **Actions** section on the Row tab.

Once the user has performed all desired edits for a particular configuration click 'OK' to save the edits and exit the dialog or 'Cancel' to exit the dialog without saving any changes. All available configurations can be accessed through the 'Severity' dropdown menu.

Flash/Beep

The Flash/Beep functionality has been added for the latest release of AWXview32. Both of these options can be activated on the **Row - Configure** dialog.



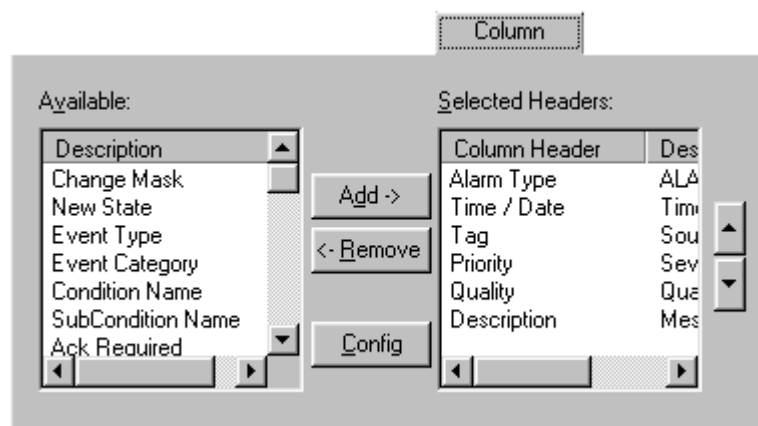
Activating Flash and Beep

Simply checking the box next to each feature will activate it and enable the 'Frequency' option: Slow, Medium or Fast.

It is possible to change the default beep to a *.wav file by accessing **Settings - Control Panel - Sounds - Default Sound**. Please note that if you chose a large file as your default sound your system may experience difficulty when trying to load it.

The Flash option will simply invert the text and background color to indicate the flashing.

Column



The Column tab allows the user to select which columns they want to appear in the Viewer. The list on the left contains all available headers, while the list on the right contains only those headers selected by the user. To add or delete headers from the right-hand list select the header and click either the 'Add ->' or '<- Remove' button.

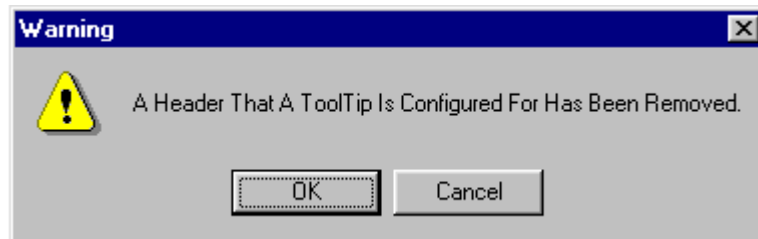
Note: Alarm Type can not be deleted from the Selected Headers list.

Note: If the user removes a header from the selected list that was used in the sort list they will be presented with the following warning message:



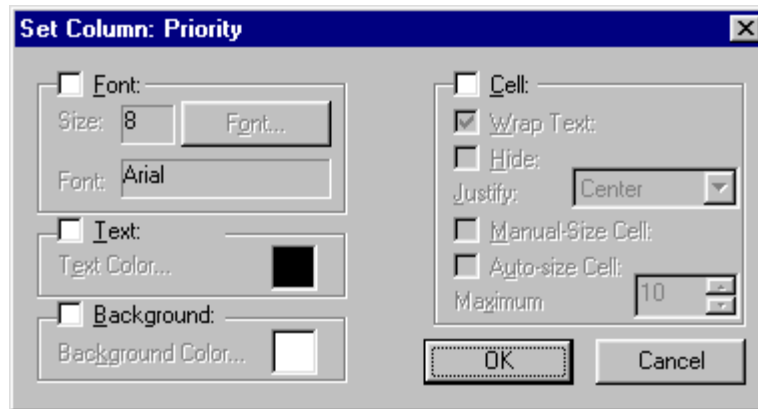
Click OK and open the Display tab to see how the Sort list has been changed.

Note: If the user removes a header from the selected list that was used in the help or comment ToolTip configuration they will be presented with the following warning message:



Click OK and open the Settings tab to see how the ToolTip configuration has been changed.

The headers listed in the 'Selected Headers' list will appear in the order that they are listed. To change this order, click the up and down arrow buttons to the right of the list until the headers are in the desired order. If the user wishes to configure the appearance of a particular header, select the header from the list on the right and click the 'Config' button. This will bring up the following dialog.



To change the configuration for Font, Column Color or Cell the appropriate box must be checked. To change the font size and style click the 'Font' button and the standard font dialog will appear. Once you have made the desired changes the new size and style will appear in the corresponding boxes.

Note: It is not possible to change the size and style without using the font dialog.

To change the Column Color click the 'Background Color' box and use the ensuing color dialog to make your changes.

To change the Text Column Color click the 'Text Color' box and use the ensuing color dialog to make your changes.

The 'Cell' section allows the user to configure how they would like the text to appear within each cell and how the size of each cell is determined. As in the 'Default' tab checking the 'Wrap Text' box makes the text wrap when it hits the end of the particular cell. When this option is not applied text will occupy one line. This may result in part of the text being clipped by the adjacent cell.

Note: The 'Wrap Text' feature will only be visible with multi-line rows.

The 'Justify' field will determine the justification (left, center or right) of the text within each row. There are three options for determining the size of each cell:

- Manual-Size Cells

Checking this box indicates that the user would like to be able to set the width of this column. Manual sizing of columns can only be done during runtime.

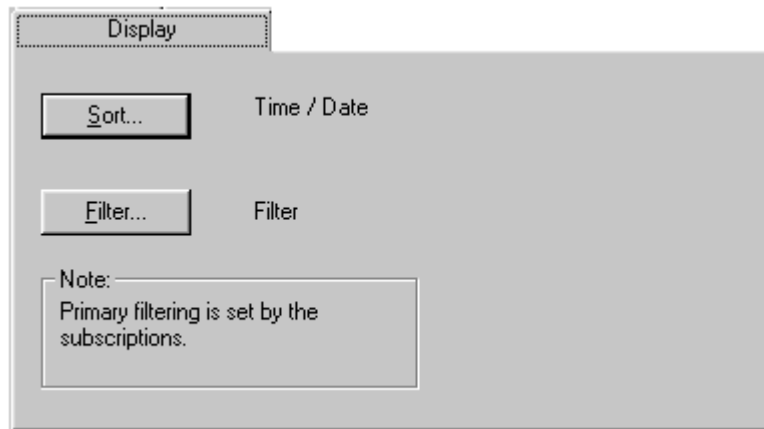
- Auto-Size Cells

Checking this box indicates that the user would like the Viewer to set the width of this column. The AlarmWorX32 Viewer will automatically set this column to the width nearest to the size of the ActiveX divided by the number of visible columns.

Entering a maximum length for each cell.

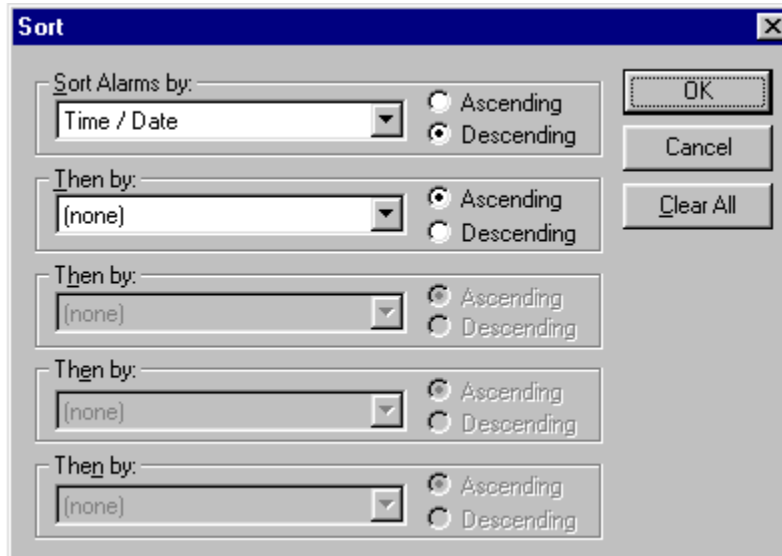
Currently this is a range from 1 to 1000 and applies the maximum cell length as the desired cell length for the selected column only. To enable this option, make sure that both Manual-Size and Auto-Size are not checked.

Display



The Display tab allows the user to configure options for sorting or client side filtering.

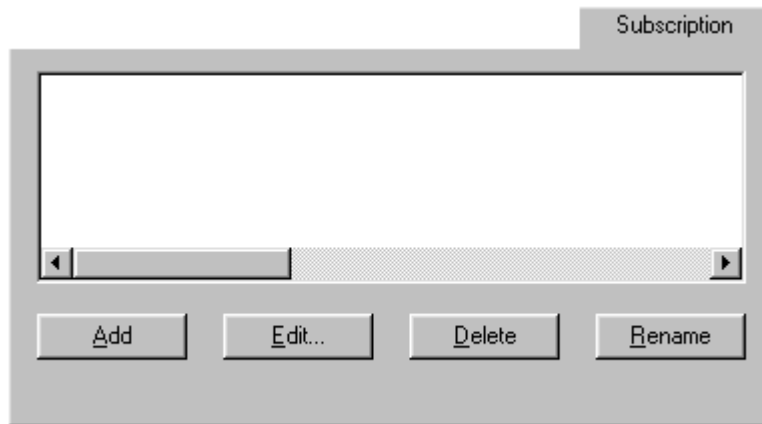
To configure the sorting feature click the 'Sort' button and the following dialog will appear.



The Sort dialog allows the user to configure which column they want to sort the alarms by. It also allows for multi-level sorting up to five levels with ascending/descending specific to each level. The AlarmView32 will always sort. The default sort for a new viewer is Time/Date

Note: This Sort dialog, including any configuration, is also available in runtime.

Subscription



The Subscription tab determines what type of OPC connection will be made with the Current Events Viewer. It allows the user to add, rename, delete, or edit a subscription to a particular alarm. To add a new subscription click the 'Add' button and the subscription named New Subscription will appear. This subscription does not contain any data so it is necessary to immediately edit the new subscription. To do so, click the 'Edit' button and the 'Event Subscription' dialog will appear with the following tabs. It is only possible to add or edit subscriptions that are connected to active OPC alarm and event servers.

Note: It is possible for a server to have more than one subscription. In fact it is a very effective way to achieve filtering.

Server
Types
Categories
Areas
Source
Attributes

Server

Server

Node: Browse...

Event Server:

The Server tab allows the user to select an Event Server and Node for each subscription. To select the event server click the 'Browse' button and the Sever Browser dialog will appear listing all available Alarm OPC Servers. Select the desired server and click 'OK'. For local servers it is not necessary to fill in the Node field.

Types

Types

Event Types:

- ☒ Simple
- ☒ Tracking
- ☒ Condition

Severity Range

Low:

High:

The Types tab allows the user to configure which OPC defined event types each subscription should have in addition to setting the ranges for severity (priority). 0 represents the low severity value and 1000 represents the high severity value. Please note that OPC alarm and event servers are required to scale severity values to the OPC ranges, (i.e. an alarm and event server that contained 2 severity ranges would convert these to 0 and 1000).

Simple: These messages state information but do not have alarm status and do not contain information on what initiated the message. Includes the following information: Source, Time, Type, EventCategory, Severity, Message, Cookie, Server specific items.

Example "FIC101, 12:0:0 1/1/99, Simple, Category1, 100, 'Shift Change', 1"

Simple messages would be similar to an event.

Tracking: These messages contain the additional information of the client, which initiated the event. Includes the following information: Source, Time, Type, EventCategory, Severity, Message, Cookie, ActorID and Server specific items.

Example "FIC101, 12:0:1 1/1/99, Tracking, Category1, 300, 'Pump pressure Set to 10 psi', 1, Station 12"

Tracking messages would be similar to event messages where the cause of the event is important. An example would be an operator changing a setpoint value. This type of message does not include acknowledge capability.

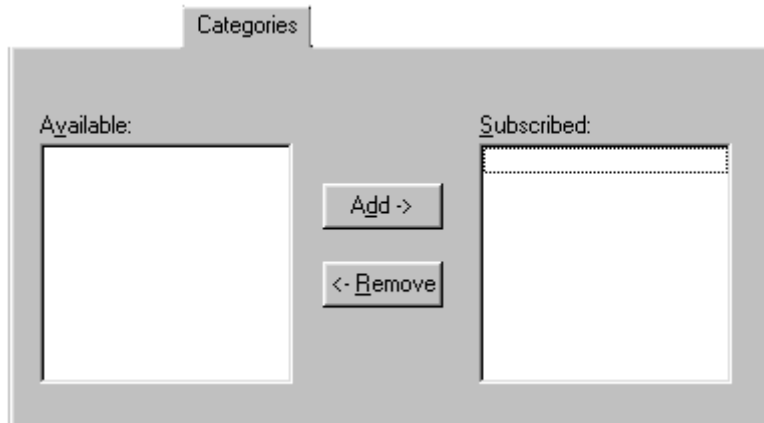
Note: Simple and Tracking messages are removed from the Alarm viewer via the acknowledge mechanism.**Condition:** These messages contain all of the above information but also include an acknowledgement portion. Includes the following information: Source, Time, Type, EventCategory, Severity, Message, Cookie, ConditionName, SubConditionName, ChangeMask, NewState, ConditionQuality, AckRequired, ActiveTime, ActorID and Server specific items.

Example "FIC101, 12:0:3 1/1/99, Condition, Category1, 700, 'Pump pressure to high', 1, Limit, HiHi, 1, Active Enabled, Good, TRUE, 12:0:2 1/1/99"

Condition messages would be considered a "typical" alarm message with acknowledge capability.

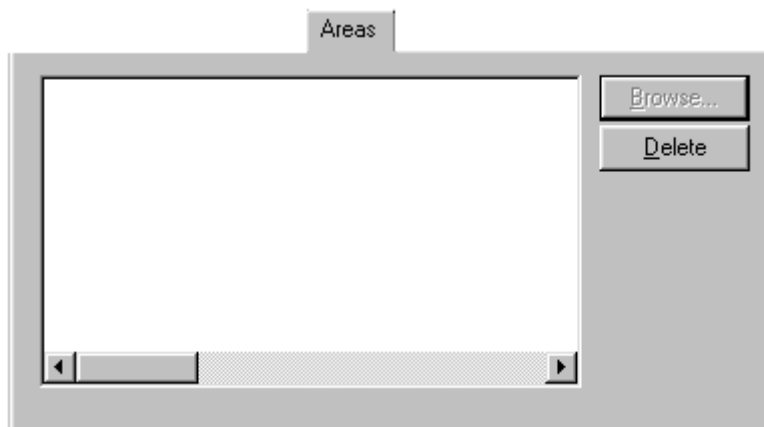
For exact details on any of the included information please see the OPC Alarm and Events Specification.

Categories



To select the categories for a subscription, select the category from the list of available categories and click the 'Add ->' button and it will appear in the Subscribed list. To remove a category from this list select it in the Subscribed list and click the '<- Remove' button. If no categories are listed in the Subscribed list then all categories are selected by default.

Areas

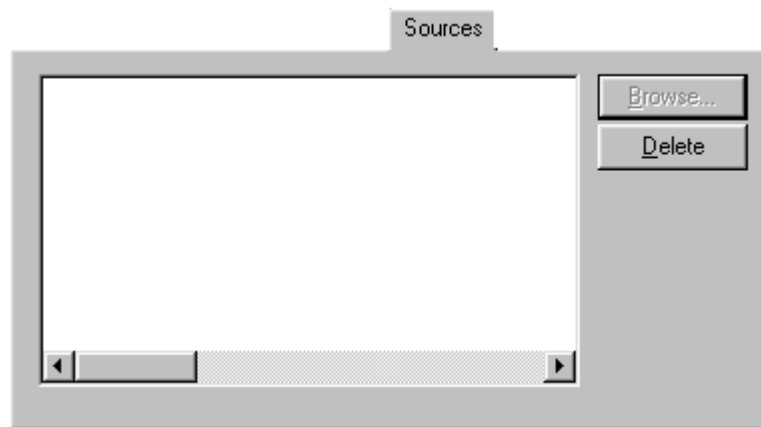


To select an area or group of areas for filtering in a subscription, click the 'Browse' button and the 'OPC Event Server Area/ Source Browser' will appear listing all available Areas for your Event Server. Area subscription supports wildcards. Wildcards subscribe to the format of Visual Basic's like command. For example, Area1* will subscribe to all alarm areas that contain strings beginning with Area1. Area1* will subscribe to the root area and its child areas. A detailed explanation of the wildcard support can be found in the OPC Alarm and Events documentation. It is a good idea to thoroughly read the wildcard documentation before attempting to use

complicated expressions. It is also possible to delete an area from this same window. To delete an area, select it from the list of areas for this particular subscription and click the 'Delete' button. Not all OPC alarm and event servers support area filtering as part of the subscription.

Note: If no areas are listed then all areas are selected by default.

Source



The Source tab allows the user to filter on a source or group of sources for a particular subscription. To select a click the 'Browse' button and select one from the OPC Event Server Area/ Source Browser. Source subscription provides the wildcard support found in the area subscription. It is also possible to delete a source for a particular subscription. To delete a source, select it from the list of sources for this particular subscription and click the 'Delete' button. Not all OPC alarm and event servers support source filtering as part of the subscription.

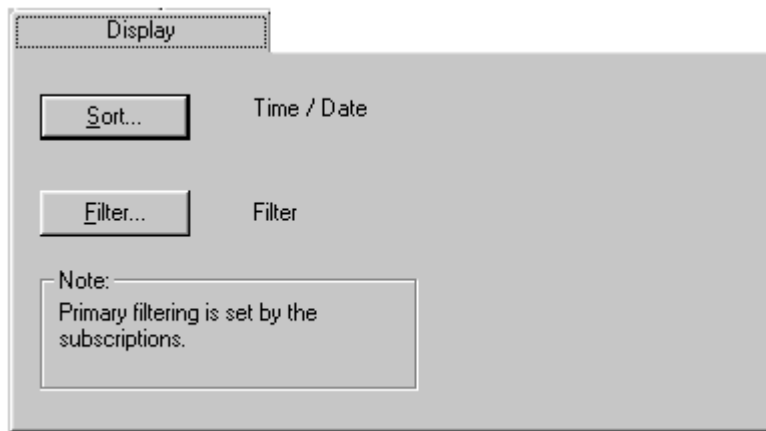
Note: If no sources are listed then all sources are selected by default.

Attributes

Client Side Filtering

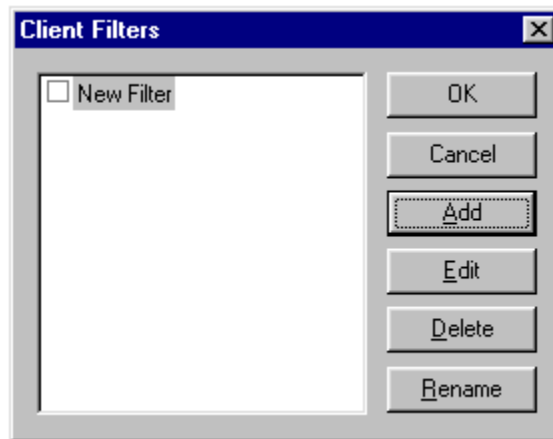
There are two ways to achieve filtering using the Alarm Viewer: Server Side and Client Side. Server Side filtering takes place through the use of subscriptions while Client Side filtering takes place through the use of the Filter option on the Display tab of the AWWview32 ActiveX Properties dialog.

Note - When possible, Server Side filtering should be the primary filtering method.



Display Tab

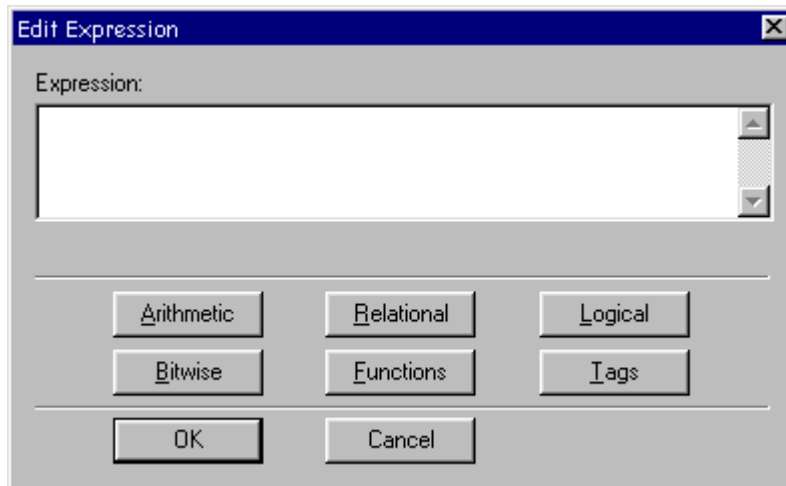
Click the Filter button the following dialog will appear.



Filter Configuration Dialog

Add/Edit

To Add a filter click the 'Add' button and a new filter will appear in the list box. To enter the criteria for the filter select it and click the 'Edit' button. The following Expression box will appear:



Filter Expression Configuration

This expression box has the following options: Arithmetic, Relational, Logical, Bitwise, Functions and Tags. Each of these options will be explained in greater detail in following text.

Delete

To delete a filter all together - select the filter and click the 'Delete' button. Your system will ask if you really want to delete the filter - click 'Yes' and the filter will be deleted.

Rename

To rename a filter, simply select the filter and click the 'Rename' button. It is necessary to change the filter name if you are going to add more than one filter. Try to give the filter a name that describes what it does or the values it considers when filtering.

Note: Depending on the security level of the user filters may or may not be able to be created or activated during runtime.

Arithmetic
Relational
Logical
Bitwise
Functions
Tags
Grids
Settings

Arithmetic

The symbols '+', '-', '*', '/' and '%' use the following format

expression :: parameter **symbol** parameter

Where

parameter a local variable, an OPC tag, a constant or another expression

symbol + or - or * or / or %

Result:

The expression results in a number of any type (float, long, etc.)

Examples:

Sym bol	Descripti on	Example	Result
+	Addition	~~var1~~ + ~~var2~~	8+3=11
-	Subtracti on	~~var1~~ - ~~var2~~	8-3=5
*	Multiplic ation	~~var1~~ * ~~var2~~	8*3=24
/	Division	~~var1~~ / ~~var2~~	8/3=2.66667
%	Calculate s the remainde r after division	~~var1~~ % ~~var2~~	8%3=2
(and)	Gives preceden ce to parts of the calculati on	~~var1~~ / (~~var2~~ + ~~var3~~)	8/(3+2)=1.6

Relational

The symbols '<','>','<=','>=','==' and '!=' use the following format

expression :: parameter **symbol** parameter

Where

parameter a local variable, an OPC tag, a constant or another expression

symbol < or > or <= or >= or == or !=

Result:

The expression results in a Boolean value (0 or 1).

Examples:

Sy mb ol	Descripti on	Example	Result
<	Smaller than	~~var1~~ < ~~var2~~	8<3 = 0
>	Greater than	~~var1~~ > ~~var2~~	8>3 = 1
<=	Smaller than or equal to	~~var1~~ <= ~~var2~~	8<=3 = 0
>=	Greater than or equal to	~~var1~~ >= ~~var2~~	8>=3 = 1
==	Equal to	~~var1~~ == ~~var2~~	8==3 = 0
!=	Not equal to	~~var1~~ != ~~var2~~	8!=3 = 1

Logical

The symbols '&&' and '||' use the following format

expression :: parameter **symbol** parameter

The symbol '!' uses the following format

expression :: **symbol** parameter

Where

parameter a local variable, an OPC tag, a constant or another expression

symbol && or || or !

Result:

The expression results in a Boolean value (0 or 1).

Truth table:

~~var1~~	0		not 0	
~~var2~~	0	not 0	0	not 0
~~var1~~ && ~~var2~~	0	0	0	1
~~var1~~ ~~var2~~	0	1	1	1
!~~var1~~	1	1	0	0

Examples:

Sym bol	Descripti on	Example	Result
&&	And	~~var1~~ && ~~var2~~	8 && 3 = 1
	Or	~~var1~~ ~~var2~~	8 3 = 1
!	Not	!~~var1~~	!8 = 0

Bitwise

The symbols '&', '|', and '^' of the bitwise group use the following format

expression :: parameter **symbol** parameter

The symbol '~' of the logical group uses the following format

expression :: **symbol** parameter

The symbols 'shl' and 'shr' of the bitwise group use the following format.

expression :: **symbol** (value, shiftby)

Where

parameter a local variable, an OPC tag, a constant or another expression

symbol && or || or ^ or shl or shr or ~

Result:

The expression results in a number when the parameters used contain numbers.

Bit table:

	Binairy (Decimal)	Binairy (Decimal)
~~var1~~	0000.0000.0000.1000 (8)	0000.0000.0110.0000 (96)
~~var2~~	0000.0000.0000.1010 (10)	0000.0000.0000.1000 (8)
~~var1~~ & ~~var2~~	0000.0000.0000.1000 (8)	0000.0000.0000.0000 (0)
~~var1~~ ~~var2~~	0000.0000.0000.1010 (10)	0000.0000.0110.1000 (104)
~~var1~~ ^ ~~var2~~	0000.0000.0000.0010 (2)	0000.0000.0110.1000 (104)
shl (~~var1~~,3)	0000.0000.0100.0000 (64)	0000.0011.0000.0000 (768)
shr (~~var1~~,3)	0000.0000.0000.0001 (1)	0000.0000.0000.1100 (12)
~(~~var1~~)	1111.1111.1111.0111 (-9)	1111.1100.1111.1111 (-97)

Examples:

Sy mb ol	Description	Example	Result
&	Bit And	<code>~~var1~~ & ~~var2~~</code>	<code>8 && 3 = 0</code>
	Bit Or	<code>~~var1~~ ~~var2~~</code>	<code>8 3 = 11</code>
^	Bit eXclusive Or	<code>~~var1~~ ^ ~~var2~~</code>	<code>8^3=11</code>
shl	Bit shift left	<code>shl(~~var1~~, 3)</code>	<code>8<<3=64</code>
shr	Bit shift right	<code>shr(~~var1~~, 3)</code>	<code>8>>3=1</code>
~	Not (two's complement)	<code>~(~~var1~~)</code>	<code>!8 = -9</code>

Functions

The symbols 'sin', 'asin', 'cos', 'acos', 'tan', 'atan', 'log', 'ln', 'exp', 'sqrt', 'abs', 'ceil', and 'floor' use the following format

expression :: **symbol** (parameter)

The symbols 'pow', 'min' and 'max' use the following format

expression :: **symbol** (parameter, parameter)

The symbol 'if' uses the following format

expression :: **symbol**
(parameter, parameter, parameter)

Where

parameter

a local variable, an OPC tag, a constant or another expression

symbol

sin or asin or cos or acos or tan or atan or log or ln or exp or sqrt or abs
or ceil or floor or min or max or pow or if

Result:

The expression results in a number.

Examples:

Sym bol	Description	Example	Result
sin	sine of an angle in radians	sin(~var1~)	sin(0.785)=0.71
cos	cosine of an angle in radians	cos(~var1~)	cos(0.785)=0.71
tan	tangent of an angle in radians	tan(~var1~)	tan(0.785)=1.0
asin	arc sine returns an angle in radians	asin(~var1~)	asin(0.5)=0.52
acos	arc cosine returns an angle in radians	acos(~var1~)	acos(0.5)=1.05
atan	arc tangent returns an angle in radians	atan(~var1~)	atan(1)=0.785
sqrt	returns the square root	sqrt(~var1~)	sqrt(100)=10
pow	returns value 1 raised to the power value 2	pow(~var1~,~var2~)	pow(100,1.5)=1000
log	10 based logarithm	log(~var1~)	log(100)=2
ln	e based logarithm	ln(~var1~)	ln(7.389)=2
exp	Exponential	exp(~var1~)	exp(2)=7.389
abs	absolute value	abs(~var1~)	abs(-1)=1
ceil	integer ceiling	ceil(~var1~)	ceil(7.39)=8

Sym bol	Description	Example	Result
floor	integer floor	floor(~~var1~~)	floor(7.39)=7
min	lowest value of two	min(~~var1~~,~~var 2~~)	min(10,5)=5
max	highest value of two	max(~~var1~~,~~var 2~~)	min(10,5)=10
if	Conditional statement	if(~~var1~~<~~var2 ~~, ~~var1~~,~~var2~~)	if(5<8,5,8)=5
like	Wildcard String Compare	like(string, pattern, casesensitive*)	

Note: For the like operator string = the string to search in, pattern = the string to search for (can include wildcards), nonzero for case-sensitive search, zero for case-insensitive search. String syntax is \$"string"\$.

Tags

When the Tags button is pushed, the following dialog will appear:



Alarm Tag dialog

Select the attribute that you want to include in the filter expression and click 'OK'.

There are two new attributes available for use in Filtering: Alarm Type and Current Time. The Alarm Type attribute will allow the user to filter alarms according to ALARM 1, ACK 2, UNACK 3, OPER 4, TRACK 5 or NORM 6. For example the user can set up a filter with the condition:

`X = {{AlarmType}}`

If the Alarm Type is true then the alarms are seen, if they are false then they are not displayed.

The Current Time attribute allows the user to filter according to the current time. Only alarms occurring around the current time will be displayed.

Example Filters

`X = {{Severity}} > 500`. Result - Only alarm messages with a severity greater than 500 will be visible.

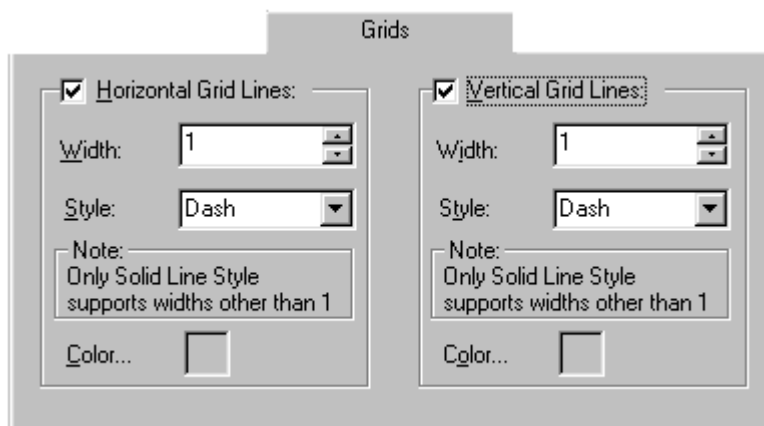
`X = Like {{Source}} , $"Tag"$,0)` Result - only messages with the tag in the source name will be displayed.

`X = 1`. Result - Filter displays all messages.

`X = 0`. Result - Filter does not display any messages.

All filters resolve to TRUE or FALSE. All non-zero values resolve to TRUE.

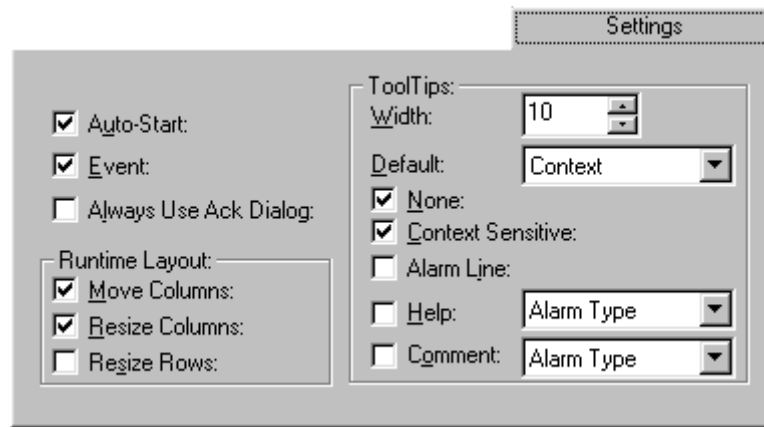
Grids



The Grids Tab pertains to the grid lines that appear on the Viewer. Grid lines will only appear if the 'Horizontal Grid Lines' or 'Vertical Grid Lines' box is checked. Once one, or both, of these boxes are checked the user is able to configure the width and style of the lines. To change the color of the grid lines simply click the color box at the bottom of the dialog and a Color dialog will appear.

Note: It is not necessary to have grid lines, but it is helpful in distinguishing between different records.

Settings



The Settings tab allows the user to configure how runtime will be started and which functions will be enabled once it has. Checking the 'Auto-Start' box will trigger the server to start runtime when the container is activated. Checking the 'Event' box allows the user automation event trigger (VB event) to be enabled during runtime. The 'Always Use Ack Dialog' forces the user to acknowledge all alarms through the Acknowledge Alarm dialog.

Runtime Layout

The Runtime Layout section allows the user to determine whether or not the order of the columns will be able to be changed in runtime, selecting a column header and moving it to the new location will achieve this in runtime. Additionally, the user can decide if columns and rows will be able to be resized while in runtime mode.

ToolTips

This section allows the user to determine which tooltip options will be available during runtime, by selecting the corresponding check boxes. The 'Default' field allows the user to select the tooltip that will be the default in

runtime, and the 'Width' field determines the width of the tooltip when it is shown in the display.

Both the 'Help' and 'Comments' fields have corresponding drop down list boxes that determine which columns will be displayed for these selections.

Runtime Operations

During Runtime, the AlarmWorX32 Viewer provides support for the following functions:

- Alarm Acknowledgement
- Sorting
- ToolTips
 - VB Event
 - Application
 - Display

Note: When the scroll bar is moved from the top position, the Alarm Viewer enters a "freeze" mode. This mode will enable a user to scroll without having the alarm sorting interfere with reading the alarm viewer. All alarms that existed before the user entered freeze mode are displayed. This includes any change of state for those alarms. If one of these alarms enters a normal state while the viewer is in freeze mode, a placeholder line will be used to maintain visual placement. All alarms that did not exist before freeze mode are processed for beep, header flash, and automation support, but are not displayed visually.

Alarm Acknowledgement
Double Acknowledging an Alarm
Sorting
ToolTips
VB Event
Display Option
Application

Alarm Acknowledgement

When an alarm is sent from the server to the viewer it will remain there until it is acknowledged in some way. The process of acknowledgement involves communication between the Alarm ActiveX (Current Alarm Viewer) and the chosen Alarm OPC Server where the alarm signals are coming from. The initial communication results in the alarm signal, which appears in the Current Alarms Viewer. Once the user acknowledges an alarm an ack message is sent to the appropriate server. The server then posts the new state of the alarm (i.e. acknowledged) to the viewer. The alarm line is then updated with the new information. Events are removed during the

acknowledgement process if included in the list of items to be acknowledged. The alarm and event servers do not receive communication for event removal. There are several ways in which an alarm or event can be acknowledged.

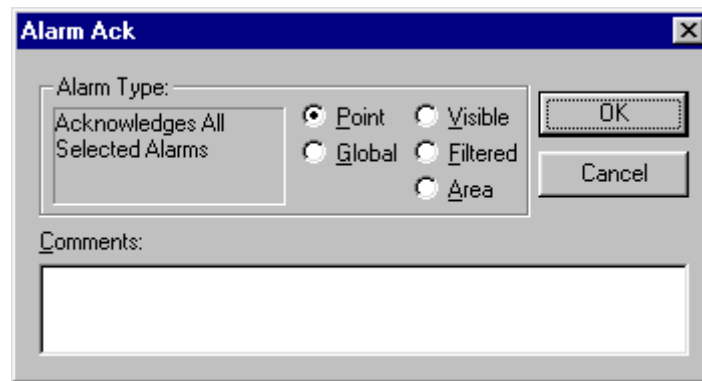
Note: The alarm header will flash whenever there is an unacknowledged alarm that is not visible in the window.

Point Acknowledgement

Point Acknowledgement will acknowledge all alarms currently selected. To select and acknowledge more than one alarm hold the control key down and left mouse click on all rows of interest, right mouse click, select point and press ok. A left double mouse click will also acknowledge all selected points if the control key is held down.

Note: The row that is under the mouse on a right click is selected if no other rows are currently selected.

- Double clicking the Alarm (if this option is selected in the Settings property page). This will acknowledge only the row double clicked on if the Ctrl key is not pressed down.
- Right clicking and selecting Ack, which brings up the following dialog



The 'Alarm Type' field allows the user to see which type of alarm they are acknowledging. Clicking the 'OK' button acknowledges a particular alarm.

Note: The Global, Visible, Filtered and Area acknowledgements will be discussed in later section.

- Another user acknowledges the alarm. In this case, the alarm will appear to automatically acknowledge.

Global Acknowledgement

When 'Global' is chosen from the 'Alarm Ack' screen the user is choosing to acknowledge all alarms that they have received up to that point. An advantage of using this option is that it will quickly clear all alarms from the current view.

Visible Acknowledgement

The 'Visible' option allows the user to acknowledge all visible alarms. Thus, if the size of the viewer allows for five alarms to be seen and a total of eight alarms are coming in, only the five visible alarms will be acknowledged. If there are more alarms than the visible ones, a scroll bar will appear on the right-hand side of the viewer.

Filtered Acknowledgement

The 'Filtered' option allows the user to acknowledge all filtered alarms. This differs from Global Alarms in the fact that Filtered Acknowledgement will not acknowledge locally filtered alarms.

Note: Only available in version 6.0 and later. In earlier versions this options behaves the same as Global Acknowledgement.

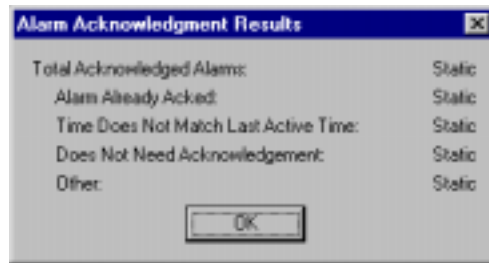
Area Acknowledgement

The 'Area' option allows the user to acknowledge all alarms that have the same area as the alarm that was selected.

Note: This option is not functional at this time.

Double Acknowledging an Alarm

It is not possible to acknowledge an alarm more than once, but since the alarm does not immediately disappear from the viewer once it is acknowledged, it is possible for a user to try to acknowledge an alarm twice. When this occurs the user will receive the following dialog box.



Total Acknowledged Alarms - provides information on the number of alarms acknowledged.

Alarm Already Acked - counts the number of alarms attempted to be acknowledged but where already acknowledged. This can occur if the alarm is acknowledged by another client before the acknowledge message is received by the client.

Time Does Not Match Last Active Time - is the number of alarms that were attempted to be acknowledged but did not contain the most recent alarm time stamp. This can occur if the alarm being acknowledged is not the most recent state of the alarm. This will occur if the acknowledge message is sent before an update of the current alarm state is received by the client.

Does Not Need Acknowledgement. - is the number of messages that did not require acknowledgement. This can occur if an acknowledgement is attempted on a non-condition message.

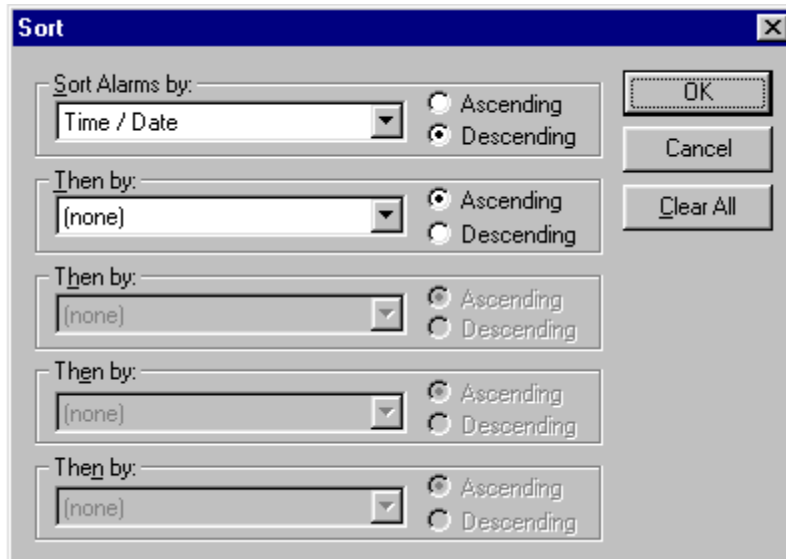
Other - is the number of messages that produced a non-OPC alarm acknowledgement COM error. Persistent occurrences of this problem could be due to a bad network connection or erroneous alarm message.

The alarm viewer will detect the current state of an alarm and will not attempt to acknowledge an alarm that is already marked acknowledged. No dialog will be posted for this occurrence.

To clear the message, simply click the 'OK' button.

Sorting

Selecting 'Sorting' from the Runtime drop down menu will bring up the following dialog.



The 'Sort Alarms by:' section indicates the first criteria the ActiveX should use when sorting all current alarms. The following 'Then by:' sections indicate how the ActiveX should sort alarms that have the same value for the criteria selected in the first section. The AlarmView32 will always sort. The default for a new control is to sort by Time/Date.

Note: It is not necessary to have any values listed in the 'Then by:' sections.

Each sorting section also has a radio button for 'Ascending' or 'Descending', which allows the user to choose which order they want to view sorted alarms in.

ToolTips

- None -No ToolTips
- Context -Context information ToolTips
- Help -Displays the information associated with the column configured for help. This is set in the Settings property page.
- Comment -Displays the information associated with the column configured for comment. This is set in the Settings property page.
- Alarm Line -Displays entire alarm line.

VB Event

This option allows the user to trigger an automation event. To use this event requires VBA programming tied to the user event.

Display Option

This option only appears if the viewer creates the display list by checking the extra attributes field for the registered file types. This attribute is server specific (for example, when using the ICONICS Alarm OPC Server select 'Default Display'). All display configured for a particular alarm will be displayed. Selecting one, by clicking on the attribute cell, will cause the display to be launched.

Application

This option only appears if the user selects the extra attribute associated with it. This attribute is server specific (for example, when using the ICONICS Alarm OPC Server select 'Default Display'). Allows the user to launch applications (i.e. *.exe, *.com or *.bat files) by clicking on the corresponding attribute cell.



Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.

Foxborough, MA 02035

Tel: 508-543-8600

FAX: 508-543-1503

E-MAIL: support@iconics.com

WEB: <http://www.iconics.com>

Contents

Introduction

Installation

Configuration

Definitions

Interface Layout

Running the Server

Introduction

The AlarmWorX32 Server receives field data from any OPC Compliant Data Access Server and performs alarm detection and reporting based on the OPC Alarm and Events Standard. The event notifications generated by the Server are sent to any OPC Alarm & Event clients that subscribe. The AlarmWorX32 Viewer and The AlarmWorX32 Logger are two examples of clients that can receive these notifications from the server.

The server supports these alarm types:

- Limit - High High, High, Low, Low Low
- Deviation - High High, High, Low, Low Low
- Rate of Change
- Digital

The Server Application (Awx32Svr.exe) has no user interface and may optionally be run as a service on Windows NT. The server reads its configuration information from a Microsoft Access Database file, which by default is **Awx32Svr.mdb** located in the same directory as the server application. During runtime the server will poll the configuration database for changes, so configuration may be done on the fly without stopping the server.

The Server Configurator Application (Awx32Cnf.exe) is used to make changes to the Microsoft Access Database file that the server uses for configuration information.

Installation

The following are the major files that comprise the server component of AlarmWorX32:

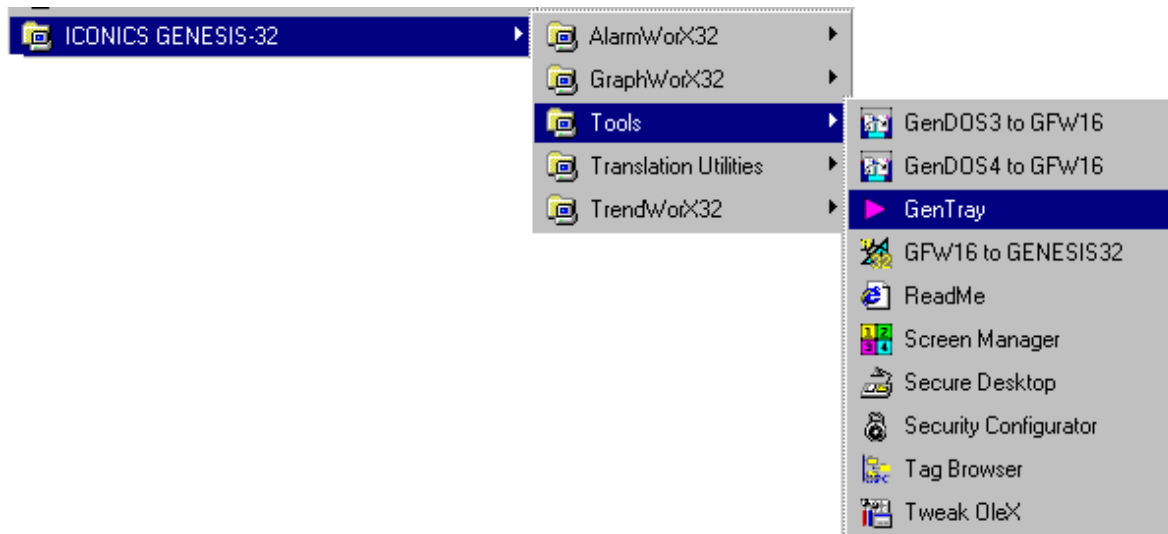
Awx32Svr.exe	Server application/service.
Awx32Cnf.exe	Server configurator.
Awx32Svr.mdb	Server Configuration Database.

The files are installed in the bin directory chosen during the install. A program icon is installed for configurator in the AlarmWorX32 program group.

By default the Server (Awx32Svr.exe) is registered as a normal application, but can be registered to run as a service under Windows NT. Running as a Service has the following advantages:

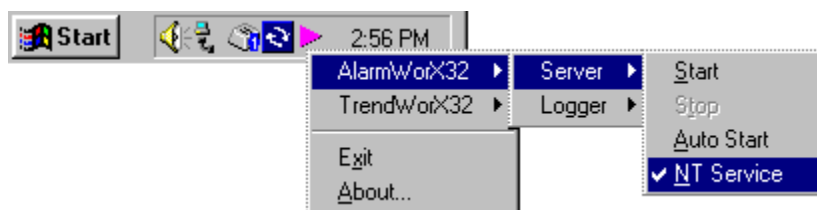
- The Server can be set to start as soon as the machine is powered (before a user has logged on).
- The Server will remain running even if a user Logs off of NT.

To register the Server as a Windows NT Service, run the GenTray utility from the Start menu by selecting **Start — ICONICS GENESIS-32 — Tools — GenTray**.



Starting GenTray

This will put the **GenTray** icon in the taskbar system tray. Click the **GenTray** icon in the system tray and select the menus **AlarmWorX32 — Server — NT Service**. To have the service start automatically at boot, click the **GenTray** icon in the system tray and select the menus **AlarmWorX32 — Server — Auto Start**.



Running GenTray

The **GenTray** utility is also used to manually start and stop the server.

Note: The Alarm Server does not use standard **COM** rules for activation/deactivation. If the server is started for any reason (via GenTray or by activation by an OPC Alarm Client), it will remain running until **both** of the following conditions are met:

1. The **GenTray** utility has issued the **Stop** command to the Server.
2. All OPC Alarm Clients have disconnected from the server.

Since NT Services are not supported in Windows 95/98, **GenTray** can be used to facilitate the starting of the Alarm Server on startup:

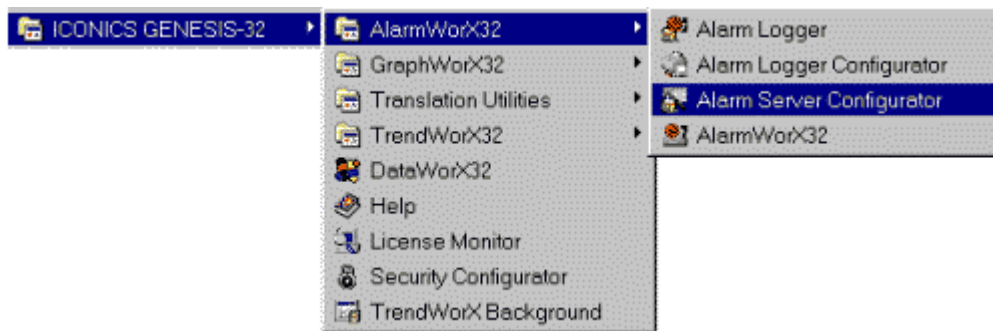
1. Place a shortcut to the **GenTray** utility in the startup group so **GenTray** will start automatically when Windows 95/98 boots.
2. To have the Alarm Server start automatically when GenTray starts, click the **GenTray** icon in the system tray and select the menus **AlarmWorX32 — Server — Auto Start**.

Configuration

AlarmWorX32 Configurator is an application used to create and maintain an alarm tag database for use by the AlarmWorX32 Server. In addition to simple tag configuration, the configurator is used to create and maintain a process area hierarchy and to group tags into one or more named "configurations" to support servers running on multiple network nodes.

To Access the AlarmWorX32 Configurator

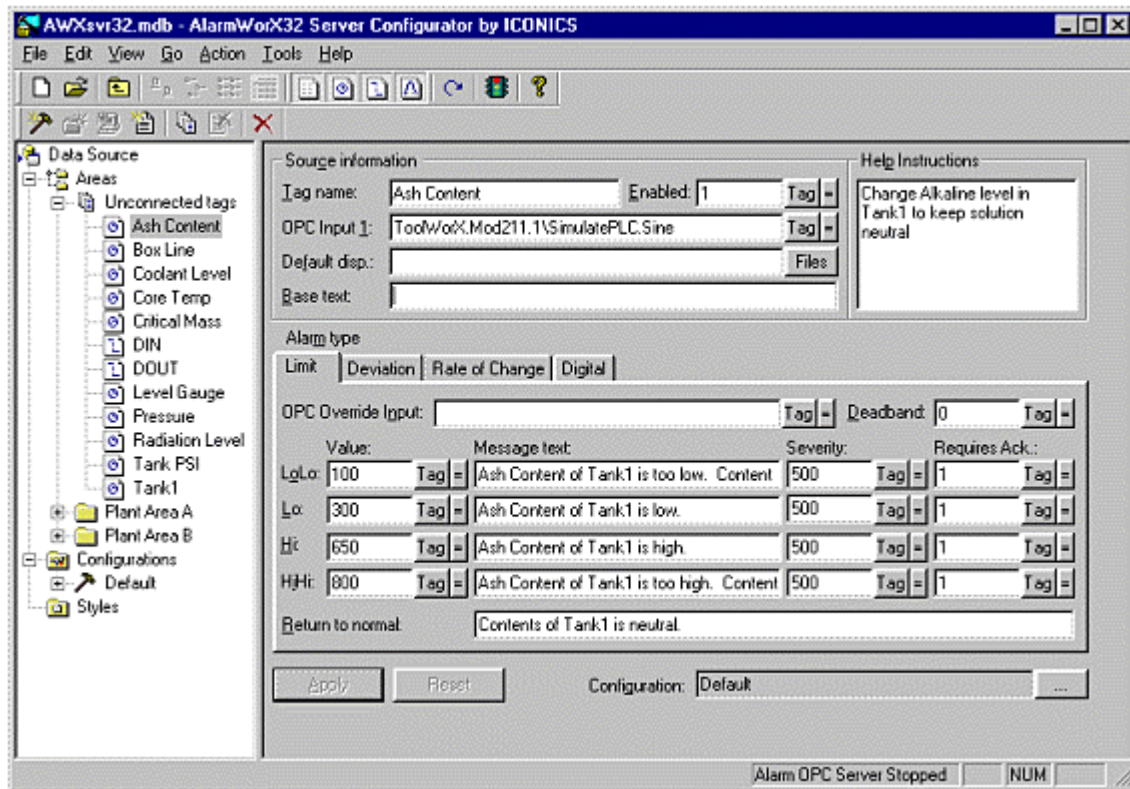
- Select **ICONICS GENESIS-32 - AlarmWorX32 - Alarm Server Configurator** from the **Start - Programs** menu.



Start - Programs menu

- The **Awx32Svr.MDB** Microsoft Access Database file will open in the Alarm Server Configurator.
- To open a different *.MDB file, select **File - Open** and the Configurator will display the 'Open' dialog box.

- To open the Configurator interface click on a listed tag and the following will display.



Definitions

Configuration

Areas

Node

Tags

Style

Configuration

A configuration is the name given to a block of information to be used for by an alarm server. Only one configuration can be active per node at one time. Only one alarm server can run per node at one time. The database can store multiple configurations.

Areas

Area is used to group OPC alarm tags. This information can be used by clients for filtering purposes. A tag can exist under multiple areas and an area can have a multiple levels. An area is defined by its name and the name of its parent branches. For example, a branch called area1 under plant area A is different than area 1 under plant area B.

Node

A node is the PC network name to associate with a particular configuration. When an alarm server starts it will check the database for its node name. If it does not find a configuration associated with its node name it will run the configuration titled 'default'. A configuration can be run on multiple nodes, however only one configuration can be run on a node (PC) at one time.

Tags

A Tag represents a single object for OPC alarm information. An alarm tag can contain information for multiple types of OPC alarms such as deviation and limit alarms. For example OPC alarm tag called FIC101 can have both deviation and rate of changes alarms associated with it.

Style

Style is a defined subset of Tag fields that are duplicated by the alarm configurator's multiply function. A database can store multiple styles for configuration.

Interface Layout

Source Information Section
Help Instructions Section
Configuration
Apply / Reset
Alarm Type Section
Menu Items

Source Information Section

The Source Information section allows the user to indicate the Tag Name, whether it is enabled or not, OPC input1, Default display and the Base Text.

Tag Name:	provides the alarm's tag name that uniquely identifies the alarm
Enabled:	uses a value, expression or OPC tag to determine if the alarm is enabled or not
OPC input1:	Indicates an expression (which can contain an OPC tag) or an OPC tag itself as the source of data for the alarm.
Default disp.:	Allows the user to choose a file to display for the selected tag.
Base Text:	Text that will appear with the alarm.

Help Instructions Section

The Help Instructions section displays user-defined Help instruction that display at the bottom of the Current Alarms screen when the associated tag reaches the alarming state to help guide operators through corrective action steps to solve a particular alarm. You can enter information directly or paste text from the Clipboard using either **Ctrl + V** or **Ctrl + Insert**.

Configuration

The configuration field is used to allow the user to associate an alarm tag with a particular configuration. Multiple configurations can be stored in a single database, however only one configuration can be active for a single alarm server at one time. This field is mandatory.

Apply / Reset

The Apply button is used to store a particular alarm tag to the database. The Reset button is used to reset the current alarm tag configuration.

Alarm Type Section

The Alarm Type section displays four tabs (Limit, Deviation, Rate of Change and Digital) that describe the limits of each tag and when an alarm will be set off.

Note: If data is entered on either the Limit or Deviation tabs the tag is defined as an Analog Tag. If data is entered on either the Rate of Change or Digital tabs the tag is defined as a Digital Tag. A tag that contains data on at least one tab from each group is classified as an Analog/Digital Tag.

Limit

OPC Override Input: Tag = Deadband: Tag =

	Value:	Message text:	Severity:	Requires Ack:
LoLo:	100 Tag =	Ash Content of Tank1 is too low. Contents are	500 Tag =	1 Tag =
Lo:	300 Tag =	Ash Content of Tank1 is low.	500 Tag =	1 Tag =
Hi:	650 Tag =	Ash Content of Tank1 is high.	500 Tag =	1 Tag =
HiHi:	800 Tag =	Ash Content of Tank1 is too high. Contents ar	500 Tag =	1 Tag =

Return to normal: Contents of Tank1 is neutral.

Alarm Type Limit tab

The limit tab allows the user to set the secondary OPC input and to set the values for the four levels of alarms: LoLo, Lo, Hi and HiHi.

The 'OPC override Input' field is used to replace the 'OPC input 1' base field for the calculation of alarms. This field is optional and if left blank the default field of 'OPC input 1' will be used.

Deadband indicates the Deadband value to apply to the converted analog values. The Deadband value is required and is calculated on borderline alarming limit values to prevent repeated alarm cycles. The 'Value' field is used to calculate the state of the input fields. For example a value of 10 for LoLo is compared against the value of OPC input 1 or OPC override Input to determine if the alarm is in LoLo state.

Severity is the OPC defined value for alarm Priority. The valid OPC severity range is 0 (lowest) to 1000 (highest).

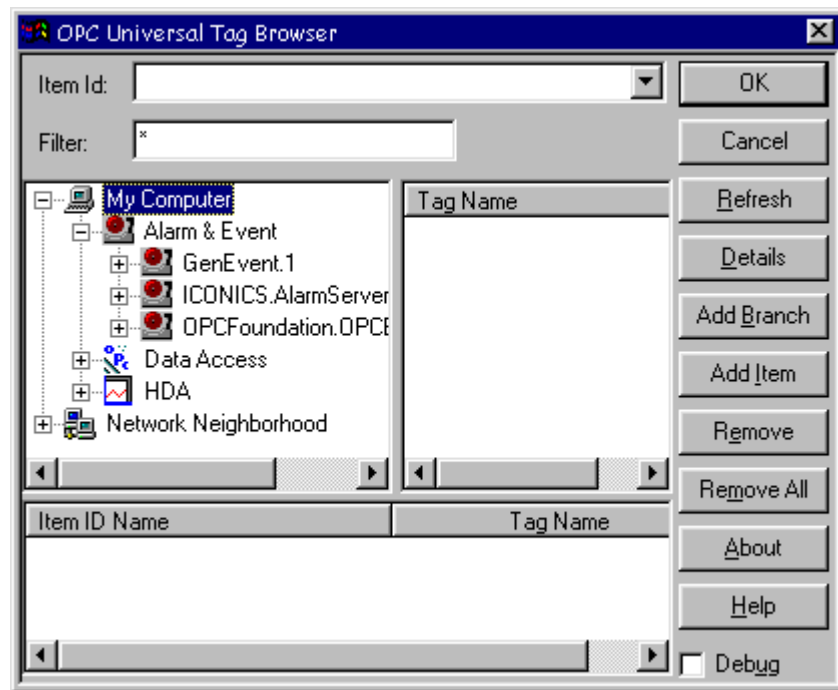
The 'Requires Ack' field is used for OPC condition alarms to determine whether the alarm needs a user acknowledgement. If the 'Requires Ack' field is one then the alarm requires a user acknowledgement. If the field is zero then the alarm is posted as already acknowledged.

Additionally, the 'Message Text' field allows the user to enter the warning message that will appear when the alarm is sent, and the 'Return to normal' field allows the user to enter the text that will appear when the alarm is taken care of.

Note: It is not necessary to enter a Message text or a base text. The Server will default to OPC subcondition name and OPC condition name. For example a lolo alarm will post a description of LOLO Limit.

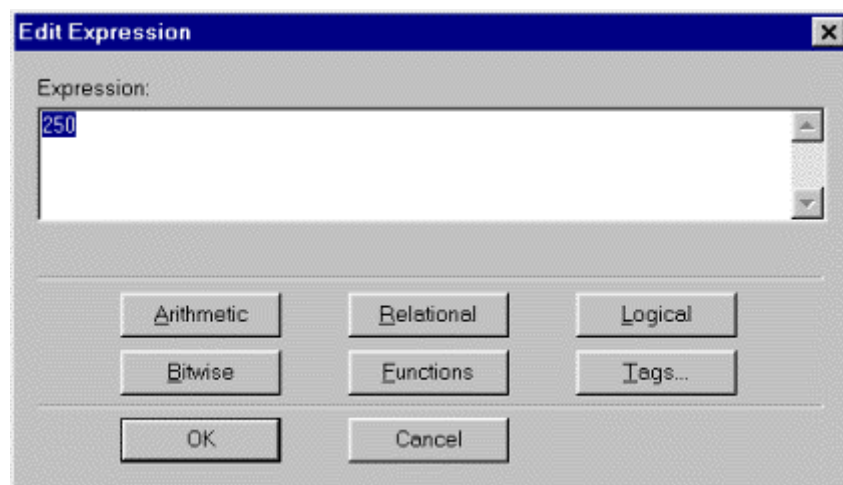
Note: Changes to the alarm property fields (HIHI, LOLO, HI, LO, Message Text, etc.) in runtime through an OPC tag update will be automatically saved to the database, over-writing any values specified in configuration mode. This is a change from AlarmWorX32 v6.0.

Several of the fields allow the user to set the value by using either the 'Tag' or '=' buttons. Using these buttons the user is able to use either the OPC Universal Tag Browser, shown below, or the expressions box, following.



OPC Universal Tag Browser

The OPC Universal Tag Browser allows the user to navigate through all installed OPC servers on their individual machine and network.



Edit Expressions Dialog

The Edit Expressions dialog can be used to set up an equation for the value of the individual alarms. The following describes the different buttons and functions available within this dialog.

Variables:

Variables can be used in expressions. How the variable needs to be referred to depends on the type of variable.

Local variables can be used inside expressions when embedded between double tildes on both sides of the variable:

~~var1~~

OPC tags can be used inside expressions when embedded between {{ and }}:

{{ICONICS.ModbusOPCServer\Dev1.Tag1}}

Arithmetic:

The symbols '+', '-', '*', '/' and '%' use the following format

expression :: parameter **symbol** parameter

Where

parameter

a local variable, an OPC tag, a constant or another expression

symbol

+ or - or * or / or %

Result:

The expression results in a number of any type (float, long, etc.)

Examples:

Sym bol	Descript ion	Example	Result
+	Addition	~~var1~~ + ~~var2~~	8+3=11
-	Subtract ion	~~var1~~ - ~~var2~~	8-3=5
*	Multiplic ation	~~var1~~ * ~~var2~~	8*3=24
/	Division	~~var1~~ / ~~var2~~	8/3=2.66667
%	Calculat es the remaind er after division	~~var1~~ % ~~var2~~	8%3=2
(and)	Gives precede nce to parts of the calculati on	~~var1~~ / (~~var2~~ + ~~var3~~)	8/(3+2)=1.6

Relational:

The symbols '<','>','<=','>','=','==' and '!=' use the following format

expression :: parameter **symbol** parameter

Where

parameter **a local variable, an OPC tag, a constant or another expression**

symbol **< or > or <= or >= or == or !=**

Result:

The expression results in a Boolean value (0 or 1).

Examples:

Sy mb ol	Descripti on	Example	Result
<	Smaller than	~~var1~~ < ~~var2~~	8<3 = 0
>	Greater than	~~var1~~ > ~~var2~~	8>3 = 1
<=	Smaller than or equal to	~~var1~~ <= ~~var2~~	8<=3 = 0
>=	Greater than or equal to	~~var1~~ >= ~~var2~~	8>=3 = 1
==	Equal to	~~var1~~ == ~~var2~~	8==3 = 0
!=	Not equal to	~~var1~~ != ~~var2~~	8!=3 = 1

Logical:

The symbols '&&' and '||' use the following format

expression :: parameter **symbol** parameter

The symbol '!' uses the following format

expression :: **symbol** parameter

Where

parameter a local variable, an OPC tag, a constant or another expression

symbol && or || or !

Result:

The expression results in a Boolean value (0 or 1).

Truth table:

~~var1~~	0		not 0	
~~var2~~	0	not 0	0	not 0
~~var1~~ && ~~var2~~	0	0	0	1
~~var1~~ ~~var2~~	0	1	1	1
!~~var1~~	1	1	0	0

Examples:

Sym bol	Descripti on	Example	Result
&&	And	~~var1~~ && ~~var2~~	8 && 3 = 1
	Or	~~var1~~ ~~var2~~	8 3 = 1
!	Not	!~~var1~~	!8 = 0

Bitwise:

The symbols '&', '|', and '^' of the bitwise group use the following format

expression :: parameter **symbol** parameter

The symbol '~' of the logical group uses the following format

expression :: **symbol** parameter

The symbols 'shl' and 'shr' of the bitwise group use the following format.

expression :: **symbol** (value, shiftby)

Where

parameter a local variable, an OPC tag, a constant or another expression

symbol && or || or ^ or shl or shr or ~

Result:

The expression results in a number when the parameters used contain numbers.

Bit table:

	Binairy (Decimal)	Binairy (Decimal)
~~var1~~	0000.0000.0000.1000 (8)	0000.0000.0110.0000 (96)
~~var2~~	0000.0000.0000.1010 (10)	0000.0000.0000.1000 (8)
~~var1~~ & ~~var2~~	0000.0000.0000.1000 (8)	0000.0000.0000.0000 (0)
~~var1~~ ~~var2~~	0000.0000.0000.1010 (10)	0000.0000.0110.1000 (104)
~~var1~~ ^ ~~var2~~	0000.0000.0000.0010 (2)	0000.0000.0110.1000 (104)
shl (~~var1~~, 3)	0000.0000.0100.0000 (64)	0000.0011.0000.0000 (768)
shr (~~var1~~, 3)	0000.0000.0000.0001 (1)	0000.0000.0000.1100 (12)
~(~~var1~~)	1111.1111.1111.0111 (-9)	1111.1100.1111.1111 (-97)

Examples:

Sy mb ol	Description	Example	Result
&	Bit And	<code>~~var1~~ & ~~var2~~</code>	<code>8 && 3 = 0</code>
	Bit Or	<code>~~var1~~ ~~var2~~</code>	<code>8 3 = 11</code>
^	Bit eXclusive Or	<code>~~var1~~ ^ ~~var2~~</code>	<code>8^3=11</code>
shl	Bit shift left	<code>shl(~~var1~~, 3)</code>	<code>8<<3=64</code>
shr	Bit shift right	<code>shr(~~var1~~, 3)</code>	<code>8>>3=1</code>
~	Not (two's complement)	<code>~(~~var1~~)</code>	<code>!8 = -9</code>

Functions:

The symbols 'sin', 'asin', 'cos', 'acos', 'tan', 'atan', 'log', 'ln', 'exp', 'sqrt', 'abs', 'ceil', and 'floor' use the following format

expression :: **symbol** (parameter)

The symbols 'pow', 'min' and 'max' use the following format

expression :: **symbol** (parameter, parameter)

The symbol 'if' uses the following format

expression :: **symbol**
(parameter, parameter, parameter)

Where

parameter

a local variable, an OPC tag, a constant or another expression

symbol

sin or asin or cos or acos or tan or atan or log or ln or exp or sqrt or abs or ceil or floor or min or max or pow or if

Result:

The expression results in a number.

Examples:

Sym bol	Description	Example	Result
sin	sine of an angle in radians	sin(~var1~)	sin(0.785)=0.71
cos	cosine of an angle in radians	cos(~var1~)	cos(0.785)=0.71
tan	tangent of an angle in radians	tan(~var1~)	tan(0.785)=1.0
asin	arc sine returns an angle in radians	asin(~var1~)	asin(0.5)=0.52
acos	arc cosine returns an angle in radians	acos(~var1~)	acos(0.5)=1.05
atan	arc tangent returns an angle in radians	atan(~var1~)	atan(1)=0.785
sqrt	returns the square root	sqrt(~var1~)	sqrt(100)=10
pow	returns value 1 raised to the power value 2	pow(~var1~,~ var2~)	pow(100,1.5)=10 00
log	10 based logarithm	log(~var1~)	log(100)=2
ln	e based logarithm	ln(~var1~)	ln(7.389)=2
exp	Exponential	exp(~var1~)	exp(2)=7.389
abs	absolute value	abs(~var1~)	abs(-1)=1
ceil	integer ceiling	ceil(~var1~)	ceil(7.39)=8
floor	integer floor	floor(~var1~)	floor(7.39)=7

Sym bol	Description	Example	Result
min	lowest value of two	min(~~var1~~, ~var2~~)	min(10,5)=5
max	highest value of two	max(~~var1~~, ~var2~~)	min(10,5)=10
if	Conditional statement	if(~~var1~~<~~var2~~, ~var1~~, ~var2~~)	if(5<8,5,8)=5

Deviation

OPC Input 2: Tag = Deadband: Tag =

	Value:	Message text:	Severity:	Requires Ack.:
LgLo:	Tag =		500 Tag =	1 Tag =
Lo:	Tag =		500 Tag =	1 Tag =
Hi:	Tag =		500 Tag =	1 Tag =
HjHi:	Tag =		500 Tag =	1 Tag =

Return to normal:

Alarm Type Deviation Tab

The 'Deviation' tab allows the configuration of alarms where the limits are tested against the difference between two inputs, 'OPC Input 1' and 'OPC Input 2'. The 'OPC input 2' field is mandatory and is used to calculate the deviation from 'OPC input 1' base field. All other fields behave in a manner consistent with the Limit tab.

Rate of Change

OPC Override Input: Tag -

ROC Limit / Sec: Tag - Message text: Severity: 500 Tag - Requires Ack.: 1 Tag -

Return to normal:

Alarm Type Rate of Change Tab

The 'Rate of Change' tab allows the user to set an alarm if an input changes at a rate higher than **Roc Limit/Sec**. The remaining fields on this tab are identical to those on the **Limit** tab.

Digital

OPC Override Input: Tag -

Alarm State Value: Tag - Message text: Severity: 500 Tag - Requires Ack.: 1 Tag -

Return to normal:

Alarm Type Digital Tab

The 'Digital' tab allows the user to set an alarm if the comparison between the 'Alarm State Value' and the input state TRUE. The remaining fields on this tab are identical to those on the **Limit** tab.

Calculation of Alarms:

It is important to note that alarm states are calculated per update of an OPC tag. This means that alarm tags that have multiple OPC tags associated with a state will be calculated once for each OPC tag update. For example: A limit alarm has an OPC tag for the 'OPC input 1' field and for the 'LoLo' field. Both tags change value and are updated to the alarm server simultaneously. The server will then calculate the limit value once for the 'OPC input 1' field and once for the 'LoLo' field. The order of fields calculated is random and determined by the order they are placed in the update queue.

In normal operation this is not a concern, however this can cause posting of alarms due to "noise" situations. Continuing with the above example, if the LoLo value changed from 5 to 15 and the OPC input value changed from 6

to 16 an alarm could be posted if the new LoLo value is calculated before the new OPC input value even though these two number changed "simultaneously". This can be especially true when using alias values, as each of the OPC data inputs will be calculated individually.

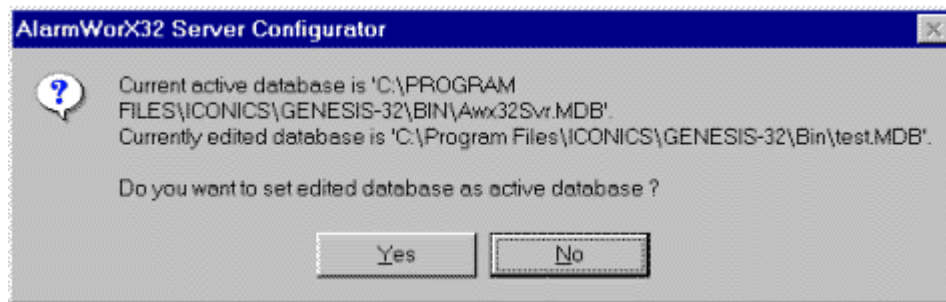
Menu Items

**File
Edit
View
Go
Action
Tools
Help**

File

The AlarmWorX32 Configurator has the following file menu options:

- New - Ctrl + N (only opens new *.mdb database files)
- Open - Ctrl + O (opens a new *.mdb (Microsoft Data Access) or *.udl (Microsoft Data Link) file.)
- Save As (allows for saving databases as *.mdb files only)
- Make Active - when choosing this option the following dialog box will appear:



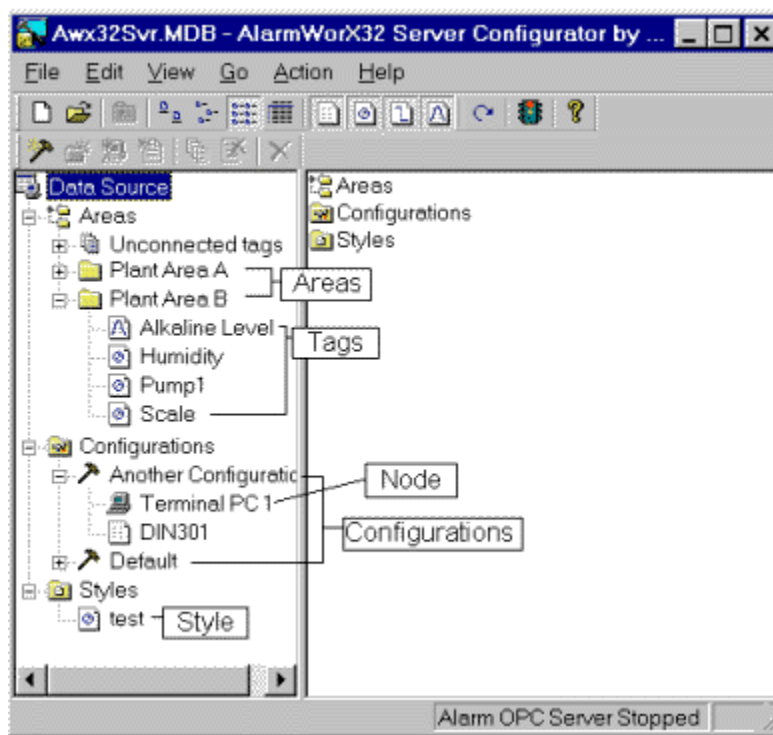
Choosing 'Yes' will make the currently edited database the active database. The active database is the one the AlarmWorX32 Server will open on start-up. The 'Make Active' option allows for easy movement from one database to another.

- Import CSV - allows the user to import data from a selected *.CSV file. Imports can be made from CSV files exported from the ICONICS Modbus Server and from CSV files exported from this configurator.
- Export CSV - allows tags and styles to be written to files in CSV format.
- List of last open MDB's (Microsoft Access Databases)
- Exit

Edit

The Edit menu contains options that allow the user to change the Configurator interface, including the following:

- Insert
 - * Configuration - Ctrl + F
 - * Area - Ctrl + A
 - * Node - Ctrl + E
 - * Tag - Ctrl + T
 - * Style - Ctrl + Y

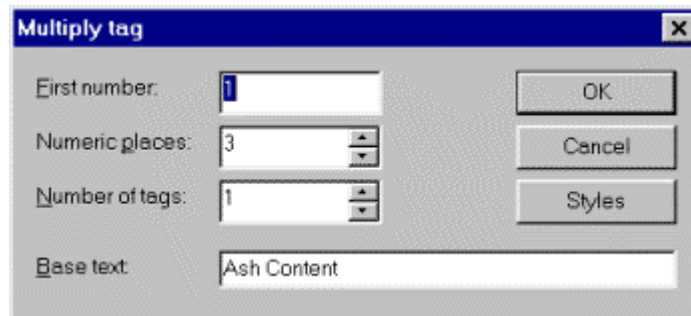


AlarmWorX32 Configurator Tree Setup

- Rename - Ctrl + R

- Multiply Tag - Ctrl + M

To multiply a tag, select the tag and then choose 'Multiply Tag' from the Edit menu and the following dialog will appear.



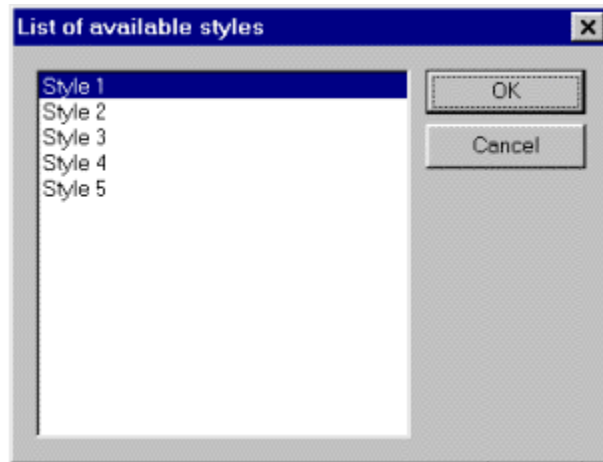
Multiply Tag Dialog

First number	Refers to the number that will be attached to the first multiplied tag
Numeric places	Refers to how many digits should be attached to the end of the initial tag name
Number of tags	Determines how many tags will be added to current configuration
Base text	Sets the initial tag name for all multiplied tags
Styles button	Brings up the 'Set Default Tag's Styles' dialog, allowing the user to change the default style for multiplied and newly inserted tags

The above dialog would produce the following tag: 'Ash Content001'

- Apply Style - Ctrl + P

When 'Apply Style' is selected from the Edit menu, or by right clicking on a tag in the Server Configurator project tree, the 'List of Available Styles' dialog will appear. From this dialog the user will be able to select from a list of pre-configured styles to apply to the current tag or group of tags selected.



List of Available Styles Dialog

Once the desired Style is selected, click 'OK' and the style will be applied.

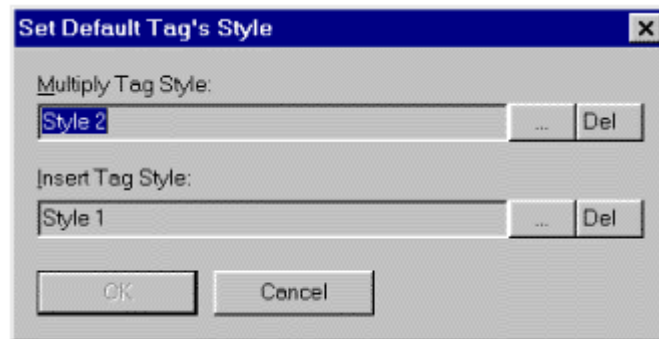
Note: The 'Apply Style' feature is useful when debugging or while running a series of tests on your system. This action is comparable to applying a template to a file, in that once the style is applied, any information contained in a field on the style will replace the tag information in that same field. If a field is left blank on the style sheet, the existing tag information in that field will remain. If a user wishes to replace information with a blank field they must enter <blank> in the style sheet.

- Delete - Ctrl + D
- Disconnect Tag(s) - Ctrl + S

Select one or more tags in an existing area and select this menu choice to disassociate the tags with this area. This will not delete the selected tags. They will appear in any other areas they are associated with or will appear under **Unconnected tags** if not associated with any other areas.

- Set Tag Styles

If the user selects 'Set Tag Styles' the following dialog will appear, allowing them to choose the default style settings.



Set Default Tag's Style Dialog

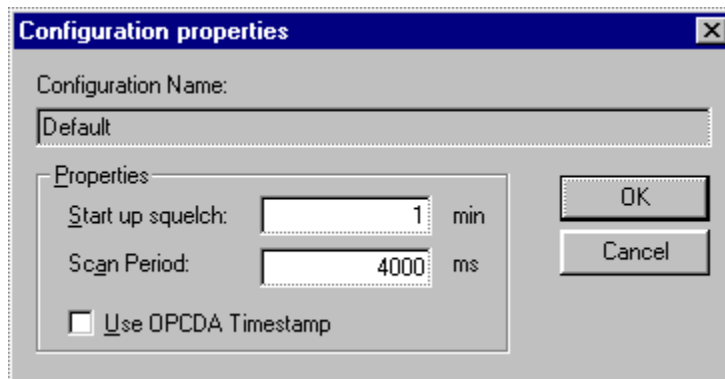
Whenever a tag is multiplied or inserted the designated style will apply. To change the default style click the '...' button and the list of available styles dialog will appear. If the user does not wish to have a default style applied when multiplying or creating a tag, simply click the 'Del' button and the value <none> will be inserted into the appropriate field. Once the Default settings are set to the appropriate values, click 'OK' to return to the main Server Configurator window.

Note: The Edit menu options are also available by left clicking in either the left-hand or right-hand panes of the AlarmWorX32 Configurator.

It is possible to move Tags and Nodes from one configuration to another, one area to another or from area to configuration and vice versa.

- Properties...

With a configuration selected, use the **Properties...** menu to edit configuration properties.



Configuration Properties Dialog

Start up squelch: The number of minutes from when the Alarm Server starts to inhibit the reporting of alarms with BAD quality. The default is one minute. When the alarm server starts, it may cause underlying OPC Data Access Servers to start. During start-up many OPC Data Access servers will send data values with BAD quality until an initial value is received from the device. Adjust this setting to eliminate erroneous alarms on startup. At the end of the defined period, any event notifications being suppressed for BAD quality will be reported.

Note: This setting does not suppress all alarm processing and reporting during this period. As soon as an input quality is first reported GOOD processing normal alarm processing begins. In most cases this happens well before the squelch period ends.

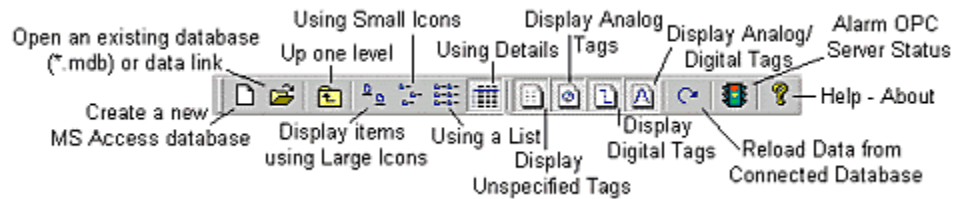
ScanPeriod: The requested update rate (period) sent to all OPC Data Access servers used by this configuration. The default is four seconds (4000 ms).

Use OPCDA Timestamp: By default the Alarm Server generates timestamps internally using the system clock. Check this box to use the timestamps provided by the OPC Data Access Servers instead. If an alarm has more than one input, the latest timestamp will be used.

View

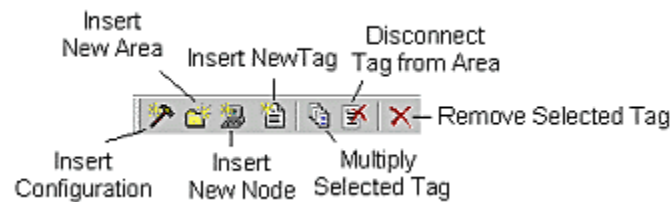
The View menu contains options that affect the view of the Configurator, including:

- Toolbar
 - * Standard Buttons



Standard Toolbar

- * Data Manipulation Buttons



Data Manipulation Buttons Toolbar

- Status Bar
- Large Icons
- Small Icons
- List
- Details

Note: It is possible to select only one of the display options (Large Icons, Small Icons, List or Details). These display options refer to how the items will be displayed in the right pane of the AlarmWorX32 Configurator.

- Sort By
 - * Name
 - * Type
 - * Date

Note: The 'Sort By' options pertain to the sorting applied to the items in the right pane of the AlarmWorX32 Configurator.

- Analog Tags
- Digital Tags
- Analog & Digital Tags
- Unspecified Tags

These checked menu items are used to show/hide alarm tags of the specified types.

- Select Language...

Allows the user to select the language to use for the configurator.

Note: User supplied text is stored to the database in one language only. Translation is not provided for user-entered text.

- Refresh - F5

Causes the configuration data to be reread from the database.

Go

The Go menu contains the following navigation aides.

- Up one level - Alt + Left Arrow
- Down on level - Alt + Right Arrow
- Next Item - Alt + Down Arrow
- Previous Item - Alt + Up Arrow
- Areas - Alt + F1
- Configurations - Alt + F2
- Styles - Alt + F3

Action

The Action menu consists of the following options:

- Start Alarm OPC Server
- Stop Alarm OPC Server

Only one of the above options will be available at a time, depending on the state of the Alarm OPC Server.

Tools

- Set working directory...

Sets the default directory used for loading and saving files. Check 'Save to Registry' to permanently change your working directory. The permanent working directory is shared by all GENESIS-32 applications.

Help

The Help menu contains:

- Help Topics
- About Application

The 'Help Topics' option will launch the On-Line help for this application. Each help file is equipped with a table of contents, index, and find tab for easy navigation. The 'About Application' option displays the ICONICS About Box containing product information for the AlarmWorX32 Server Configurator.

Running the Server

Startup
OPC Alarm and Events Interface
OPC Data Access Interface
Error Reporting

Startup

The AlarmWorX32 Server may be launched in one of four ways:

1. Automatically on boot if it is installed as an NT Service (See section 4.2 Installation)
2. By selecting 'Action', 'Start Alarm OPC Server' from the AlarmWorX32 Server Configurator menu.
3. When an OPC Alarm & Events Client creates a connection to it via COM.
4. Through GenTray.

Of these four methods, the third item is the least useful since the server should normally be running and processing alarms even if no alarm clients are active.

When launched, the Server will open the active configuration database and query for a configuration that is set to run on that node. If a configuration with a matching node name is not found, the configuration named 'default' will be run.

While running, the server will continue to poll the configuration database (every 30 seconds) in order to respond to any changes made by the Configurator.

The alarm server supports online configuration changes to allow minor changes in alarm configuration to be performed without shutting down the server. Minor things would be adding or removing a tag, changing an alarm limit, adding (but not removing) an area. Making even these minor changes online can have side effects, like duplicate events sent to clients and client messages that can no longer be acknowledged. In a live functioning plant it is possible that these minor side effects would not outweigh the alternative of shutting down and restarting the alarm server.

Whenever practical, it is desirable to toggle the server off and on to be sure only the latest configuration is being used. For major database changes

within the active database or changing the active database, the server must be shutdown and restarted.

OPC Alarm and Events Interface

The AlarmWorX32 Server supports all of the mandatory and most of the optional interfaces and methods of the OPC Alarm & Events Standard version 1.0

The following optional methods are **not** currently supported:

- IOPCEventServer::TranslateToItemIDs
- IOPCEventServer::EnableConditionByArea
- IOPCEventServer::EnableConditionBySource
- IOPCEventServer::DisableConditionByArea
- IOPCEventServer::DisableConditionBySource

All events generated are of the OPC defined Event Type 'condition'. The Server defines the following Event Categories and uses the identical names for the Condition Names:

- Deviation
- Digital
- Limit
- Rate of Change

The 'Limit' and 'Deviation' conditions have the following Sub-conditions:

- HIHI
- HI
- LO
- LOLO

The following event attributes are made available to subscribing clients:

Attribute	Dev	Dig	Lim	Roc	Type	Comment
ACK COMMENT	✓	✓	✓	✓	VT_BSTR	Comment entered when alarm was last acknowledged.
ALARM COUNT	✓	✓	✓	✓	VT_I4	Number of alarm transitions before acknowledge.
AREA	✓	✓	✓	✓	VT_ARRAY VT_BSTR	Array of areas
CV	✓	✓	✓	✓	VT_R8 VT_BOOLEAN	Current Value (OPC Input 1 or 2).
DEADBAND	✓		✓	✓	VT_R8	Alarm Detection Deadband.
DEFAULT DISPLAY	✓	✓	✓	✓	VT_BSTR	A text string, usually a file name, that when launched provides related information.
HELP TEXT	✓	✓	✓	✓	VT_BSTR	Help information
LIMIT VALUE EXCEEDED	✓		✓	✓	VT_R8	The Current Value at the time the limit was exceeded.
LOOP DESC	✓	✓	✓	✓	VT_BSTR	A description of the Source (tag).
NEXT LIM	✓		✓		VT_R8	The value of the next limit.
NORMAL STATE		✓			VT_BOOLEAN	Value of the normal state (TRUE or FALSE).
PREV LIM	✓		✓		VT_R8	Value of the previous limit reached.

SP	✓				VT_SP	Setpoint (OPC Input 2)
----	---	--	--	--	-------	------------------------

Event Attributes

OPC Data Access Interface

The AlarmWorX32 Server exposes much of its real-time and configuration data via the OPC Data Access 1.0a and 2.0 Standard. This allows OPC Data Access client applications such as GraphWorX32 to make use of alarm related data without using the OPC Alarm & Events interface. Some typical uses would include:

- Changing the color of a graphic symbol to correspond to alarm state.
- Displaying alarm limits.
- Allowing alarm limits to be changed by the operator.

The following table indicates the OPC item names that are available from the server:

Item Name	Type	R/ W	Comment
Active	VT_BOOL	R	Tag's Global Alarm Active Bit.
Name	VT_BSTR	R	Tag Name (Source).
Input1	VT_BSTR	R	
BaseText	VT_BSTR	R	
Enabled	VT_BOOL	R	Enabled state of the Source (tag).
HelpText	VT_BSTR	R	
LIM_RTNText	VT_BSTR	R	
LIM_Input2	VT_R8	R	
LIM_Deadband	VT_R8	R	
LIM_Active	VT_BOOL	R	

Item Name	Type	R/ W	Comment
LIM_Acked	VT_BOOL	R/W	
LIM_HIMI_Active	VT_BOOL	R	
LIM_HIMI_Enabled	VT_BOOL	R/W	
LIM_HIMI_RequiresAck	VT_BOOL	R/W	
LIM_HIMI_Severity	VT_R8	R/W	
LIM_HIMI_Limit	VT_R8	R/W	
LIM_HIMI_MsgText	VT_BSTR	R/W	
LIM_HI_Active	VT_BOOL	R	
LIM_HI_Enabled	VT_BOOL	R/W	
LIM_HI_RequiresAck	VT_BOOL	R/W	
LIM_HI_Severity	VT_R8	R/W	
LIM_HI_Limit	VT_R8	R/W	
LIM_HI_MsgText	VT_BSTR	R/W	
LIM_LOLO_Active	VT_BOOL	R	
LIM_LOLO_Enabled	VT_BOOL	R/W	
LIM_LOLO_RequiresAck	VT_BOOL	R/W	
LIM_LOLO_Severity	VT_R8	R/W	
LIM_LOLO_Limit	VT_R8	R/W	
LIM_LOLO_MsgText	VT_BSTR	R/W	
LIM_LO_Active	VT_BOOL	R	
LIM_LO_Enabled	VT_BOOL	R/W	
LIM_LO_RequiresAck	VT_BOOL	R/W	
LIM_LO_Severity	VT_R8	R/W	
LIM_LO_Limit	VT_R8	R/W	
LIM_LO_MsgText	VT_BSTR	R/W	
DEV_RTNTText	VT_BSTR	R	
DEV_Input2	VT_R8	R	
DEV_Deadband	VT_R8	R	
DEV_Active	VT_BOOL	R	
DEV_Acked	VT_BOOL	R/W	
DEV_HIMI_Enabled	VT_BOOL	R/W	
DEV_HIMI_RequiresAck	VT_BOOL	R/W	
DEV_HIMI_Severity	VT_R8	R/W	
DEV_HIMI_Limit	VT_R8	R/W	
DEV_HIMI_MsgText	VT_BSTR	R/W	
DEV_HI_Enabled	VT_BOOL	R/W	
DEV_HI_RequiresAck	VT_BOOL	R/W	
DEV_HI_Severity	VT_R8	R/W	

Item Name	Type	R/ W	Comment
DEV_HI_Limit	VT_R8	R/W	
DEV_HI_MsgText	VT_BSTR	R/W	
DEV_LOLO_Enabled	VT_BOOL	R/W	
DEV_LOLO_RequiresAck	VT_BOOL	R/W	
DEV_LOLO_Severity	VT_R8	R/W	
DEV_LOLO_Limit	VT_R8	R/W	
DEV_LOLO_MsgText	VT_BSTR	R/W	
DEV_LO_Enabled	VT_BOOL	R/W	
DEV_LO_RequiresAck	VT_BOOL	R/W	
DEV_LO_Severity	VT_R8	R/W	
DEV_LO_Limit	VT_R8	R/W	
DEV_LO_MsgText	VT_BSTR	R/W	
DIG_RTNTText	VT_R8	R	
DIG_Input2	VT_R8	R	
DIG_Active	VT_BOOL	R	
DIG_Acked	VT_BOOL	R/W	
DIG_Enabled	VT_BOOL	R/W	
DIG_RequiresAck	VT_BOOL	R/W	
DIG_Severity	VT_R8	R/W	
DIG_Limit	VT_R8	R/W	
DIG_MsgText	VT_BSTR	R/W	
ROC_RTNTText	VT_BSTR	R	
ROC_Input2	VT_R8	R	
ROC_Active	VT_BOOL	R	
ROC_Acked	VT_BOOL	R/W	
ROC_Enabled	VT_BOOL	R/W	
ROC_RequiresAck	VT_BOOL	R/W	
ROC_Severity	VT_R8	R/W	
ROC_Limit	VT_R8	R/W	
ROC_MsgText	VT_BSTR	R/W	
LIM_HIHI_Active	VT_BOOL	R	
LIM_HI_Active	VT_BOOL	R	
LIM_LOLO_Active	VT_BOOL	R	
LIM_LO_Active	VT_BOOL	R	
DEV_HIHI_Active	VT_BOOL	R	
DEV_HI_Active	VT_BOOL	R	
DEV_LOLO_Active	VT_BOOL	R	
DEV_LO_Active	VT_BOOL	R	

OPC Data Access Items

The values in the table indicated as Read/Write (R/W) can be changed via the OPC Data Access Interface. Any changes made to the server this way are transient, and not stored to the database. If the server is shut down these values will be lost.

Error Reporting

Internal errors encountered by the AlarmWorX32 Sever are reported via the ICONICS Event Server, another OPC Alarm & Events compliant server. The ICONICS Event Server can optionally write selected events to the NT Event Log. The following simple events may be generated by the AlarmWorX32 Server:

Severity	Message
500	"%1 %2 Started."
500	"%1 %2 Stopped."
700	"Alarm Tag: %1, Condition: %2, Input Item: %3, Error: Unknown"
700	"Alarm Tag: %1, Condition: %2, Input Item: %3, Error: Tag does not exist."
700	"Alarm Tag: %1, Condition: %2, Input Item: %3, Error: Rejected by Security Server."
700	"Alarm Tag: %1, Condition: %2, Input Item: %3, Error: Licensed I/O Point Count Exceeded. "
700	"Alarm Tag: %1, Condition: %2, Input Item: %3, Error: Demo mode has timed out."
700	"Alarm Tag: %1, Condition: %2, Input Item: %3, Error: Local I/O not allowed on Browser node."
700	"Alarm Tag: %1, Condition: %2, Input Item: %3, Error: COM error registering with GenRegistrar."
800	"%1 %2 Failed to Register with GenRegistrar. Reason: %3"
800	"No configuration found for node and no ""Default"" configuration in Database Provider: %1 Connection: %2"
800	"Database Error - Provider: %1 Connection: %2"

Server Errors

For the ICONICS Event Server, severity values of 600 and below are considered "normal". Severity values of 601 to 1000 are considered errors.



Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.

Foxborough, MA 02035

Tel: 508-543-8600

FAX: 508-543-1503

E-MAIL: support@iconics.com

WEB: <http://www.iconics.com>

Contents

Introduction

Installation

Configuration

Running the Logger

Introduction

The AlarmWorX32 Logger provides a permanent copy of alarm and event notifications produced by any OPC Alarm & Events Server, including the AlarmWorX32 Server. The alarm and event data may be stored in an OLE-DB or ODBC database and/or sent to a printer. The logger has been tested with the Microsoft Access (Jet) database and with Microsoft SQL Server.

The Logger Application (AWXLog32.exe) that provides the runtime storage and printing has no user interface and may optionally be run as a service on Windows NT. The logger reads its configuration information from a Microsoft Access Database file.

The Logger Configurator (AWXLogCfg.exe) is used to make changes to the Microsoft Access Database file that the Logger uses for configuration information.

Installation

The following three files comprise the logger component of AlarmWorX32:

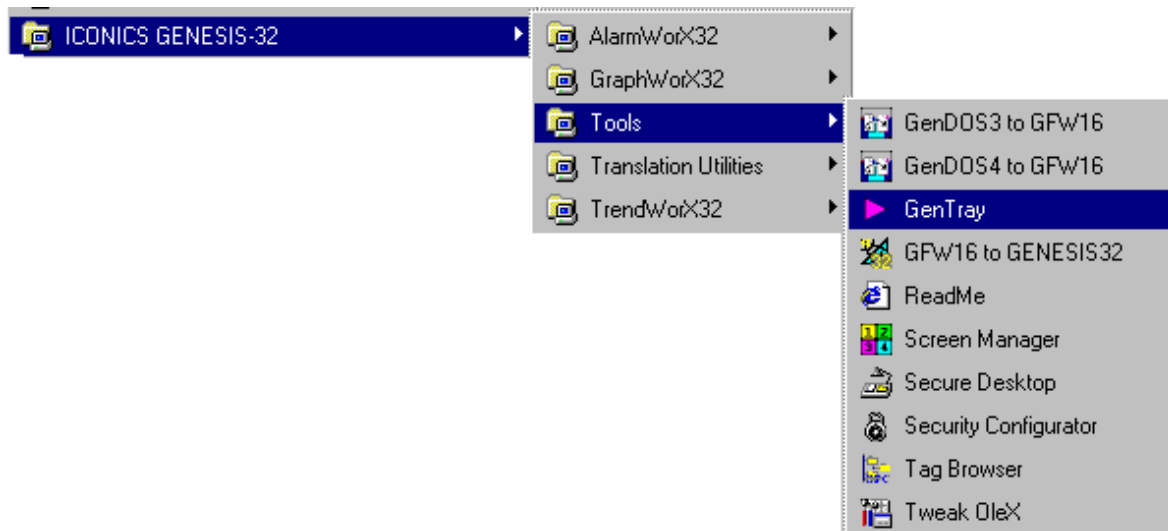
AWXLog32.exe	Logger application/service.
AWXLogCfg.exe	Logger Configurator.
AWXLog32.mdb	Logger Sample Configuration Database.

The files are installed in the **bin** directory chosen during the install. Program icons are installed for the Configurator in the AlarmWorX32 program group.

By default the Logger(AwxLog32.exe) is registered as a normal application, but can be registered to run as a service under Windows NT. Running as a Service has the following advantages:

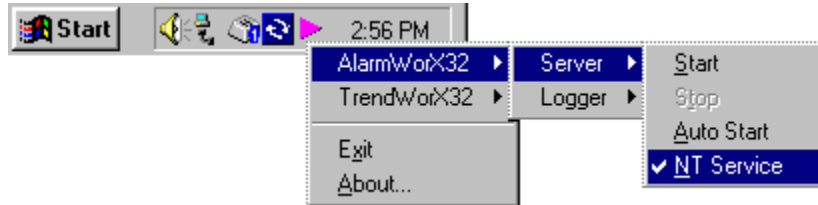
- The Logger can be set to start as soon as the machine is powered (before a user has logged on).
- The Logger will remain running even if a user Logs off of NT.

To register the Logger as a Windows NT Service, run the GenTray utility from the Start menu by selecting **Start — ICONICS GENESIS-32 — Tools — GenTray**.



Starting GenTray

This will put the **GenTray** icon in the taskbar system tray. Click the **GenTray** icon in the system tray and select the menus **AlarmWorX32 — Logger — NT Service**. To have the service start automatically at boot, click the **GenTray** icon in the system tray and select the menus **AlarmWorX32 — Logger — Auto Start**.



Running GenTray

The **GenTray** utility is also used to manually start and stop the logger.

Note: The Alarm Logger does not use standard **COM** rules for activation/deactivation. If the logger is started for any reason (via GenTray or by activation by a COM Client), it will remain running until **both** of the following conditions are met:

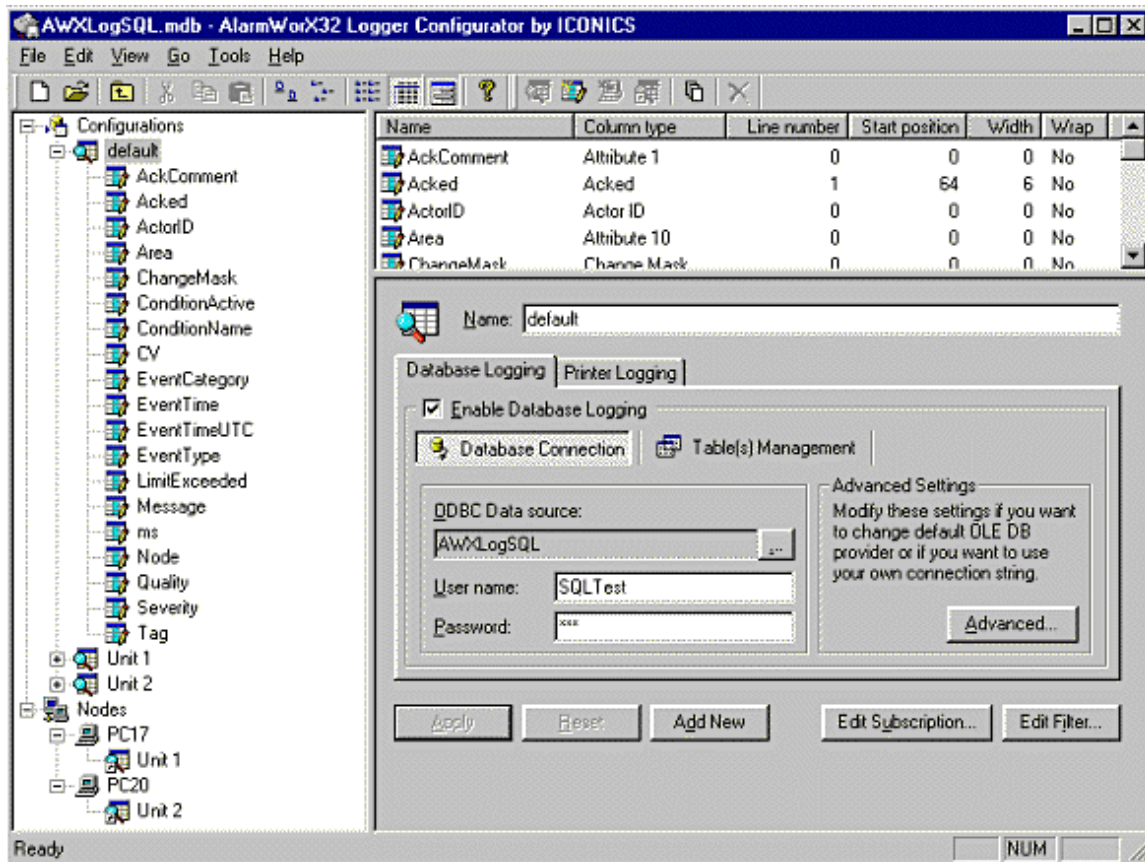
3. The **GenTray** utility has issued the **Stop** command to the Logger.
4. All COM Clients have disconnected from the logger.

Since NT Services are not supported in Windows 95/98, **GenTray** can be used to facilitate the starting of the Alarm Logger on startup:

3. Place a shortcut to the **GenTray** utility in the startup group so **GenTray** will start automatically when Windows 95/98 boots.
4. To have the Alarm Logger start automatically when GenTray starts, click the **GenTray** icon in the system tray and select the menus **AlarmWorX32 — Logger — Auto Start**.

Configuration

The AlarmWorX32 Logger Configurator writes configuration information to a Microsoft Access Database. This configuration information is read by the AlarmWorX32 Logger during runtime to determine what alarm and events to log and where to log them. A single Microsoft Access Database file (**mdb** file) can contain multiple named configurations to be run by one or more nodes on a network. A single node can also run multiple configurations simultaneously.



AlarmWorX32 Logger Configurator

The Configurator uses the familiar Windows Explorer metaphor with a hierarchical tree control on the left and a combination of list and form views on the right relating to the selected item in the tree.

Configurations
Database Logging Tab
Printer Logging
Translation Tab
Columns

Nodes
Assigned Configurations

Configurations

The Configurator Database contains one or more Logger Configurations. Each Configuration specifies a subset of alarms and events to be logged to a single database table and/or printer. Additionally the configuration defines which data fields (columns) available are to be stored and/or printed.

To insert a new configuration, Select "Configurations" from the root of the tree control and select **Edit - New - Configuration** from the menu.

The following fields, buttons and tabs on the form view are used to edit a configuration.

Note: If Dialog View is not selected from the View menu, you will not see the form!

Name

Enter a name for the configuration. The name must be unique.

Apply Button

Commits changes made in the form to the database.

Reset Button

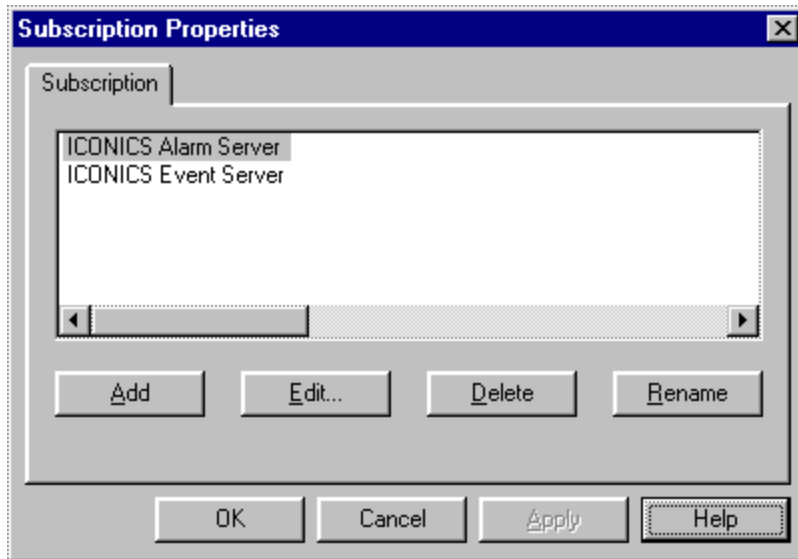
Re-reads data from the database, overwriting changes made to the form that have not yet been committed with the Apply Button.

Add New Button

Adds a new configuration to the database. Same as **Edit - New - Configuration** from the main menu.

Edit Subscription Button

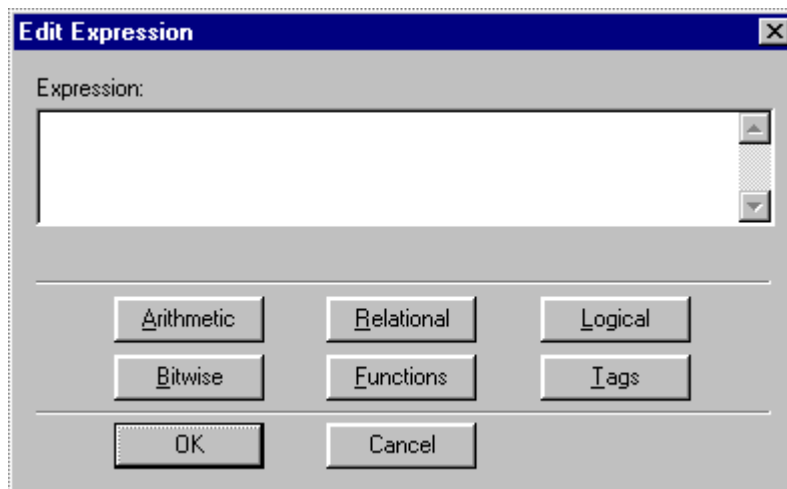
Invokes the Subscription Properties dialog box. Use this dialog to subscribe to one or more OPC Alarm & Events Server and select the desired additional attributes to be included for logging and/or printing. This process is identical to the **AlarmWorX32 Viewer**.



Subscription Properties Dialog

Edit Filter Button

Invokes the Edit Expression Dialog Box. Use this dialog to further filter the alarms and events that you want to log and/or print. This process is identical to the **AlarmWorX32 Viewer**.



Edit Expression Dialog

Database Logging Tab

Enable Database Logging Checkbox

If this box is checked, logging to the specified database will occur.

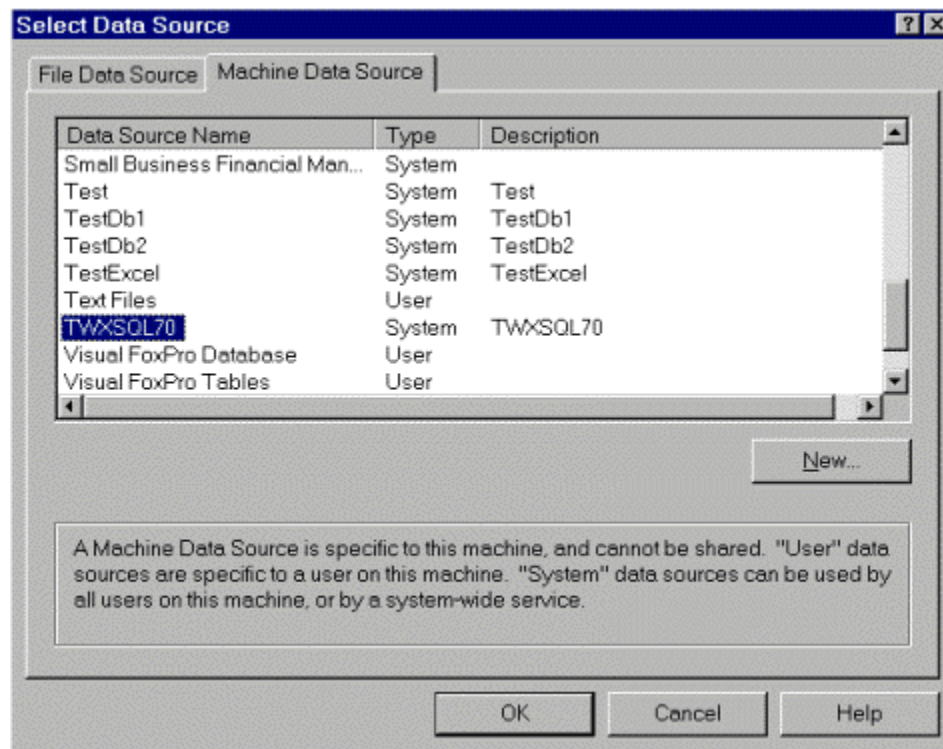
Database Connection Table Management

Database Connection

ODBC Data Source

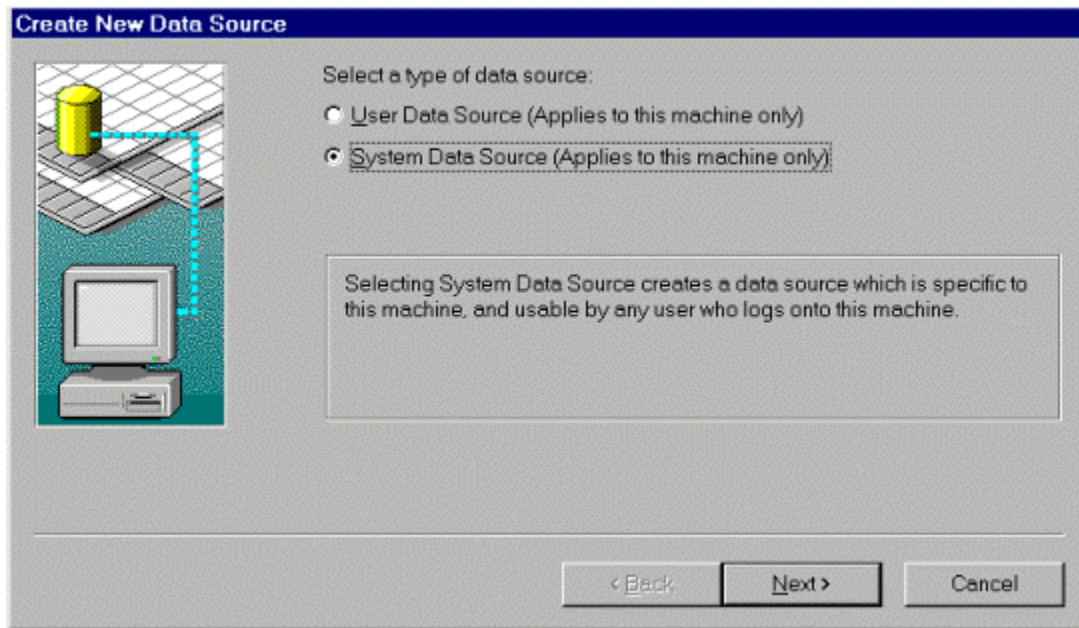
Indicates the database to use for logging. If this field is blank (the default), logging will take place to the configuration database. It is highly recommended that you change the default and log data to a different database than the configuration database.

To specify a new or existing database to log to, select the button at the end of the field. The database connection dialog is displayed.



Database Connection Selection / Creation Dialog

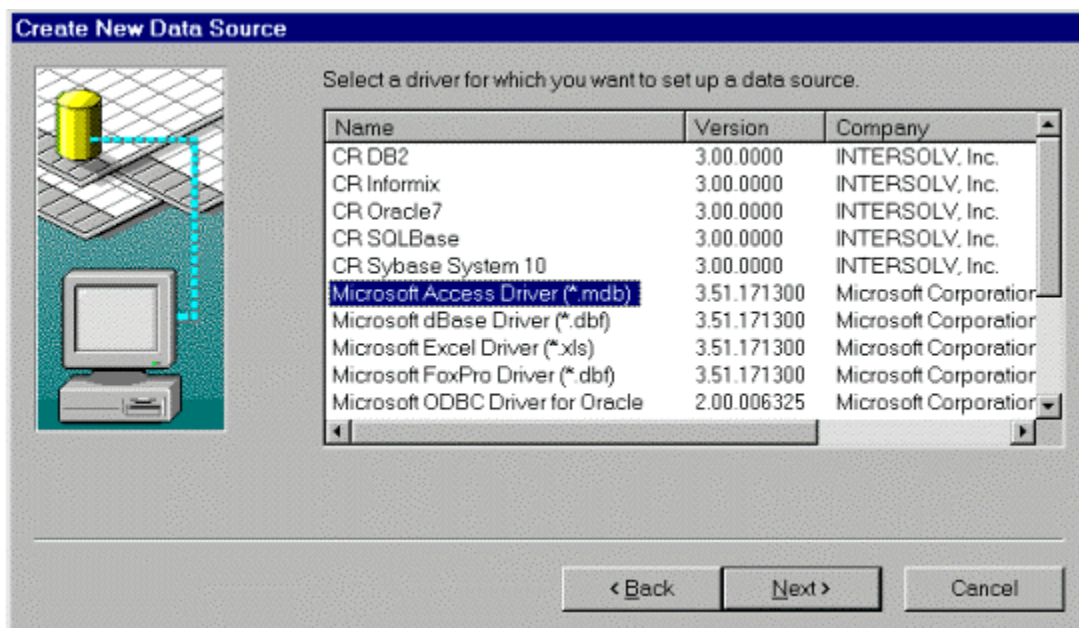
Choose the Machine Data Source Tab and either select an existing database or create a new one by clicking on New to start the configuration of a new database.



Creating a New Database Connection Dialog

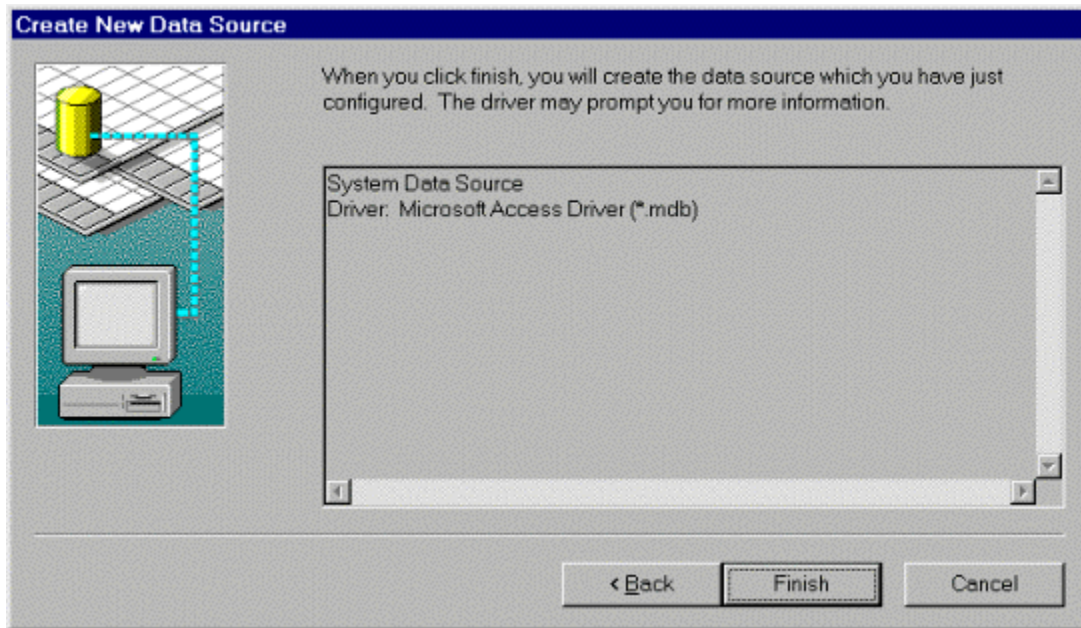
We recommend configuring the new database as a System Data Source so it is available to any user logged on the specific PC station. Next pick the corresponding ODBC driver such as Access or SQL Server.

Configuring MS Access Databases



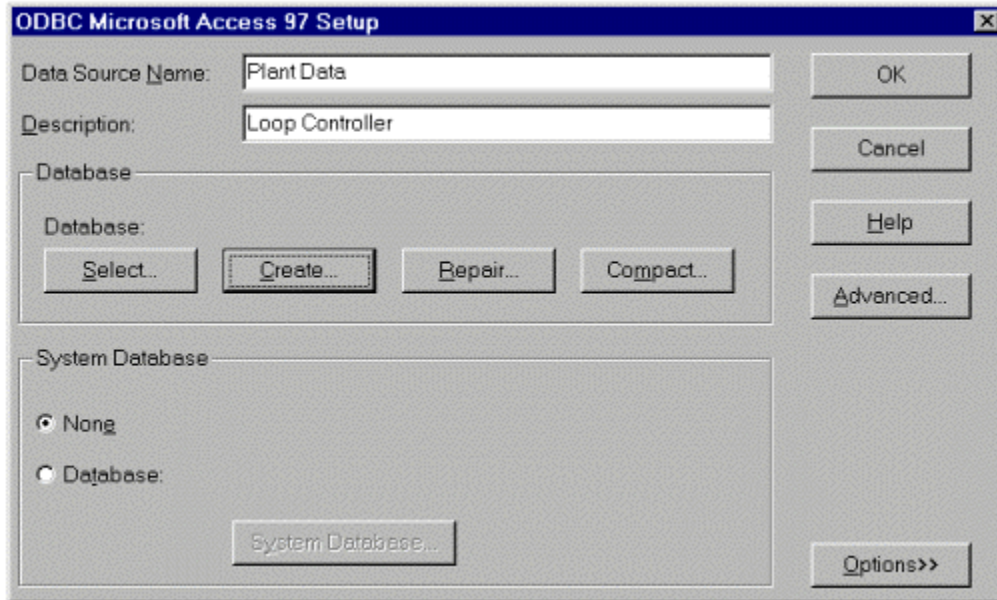
Selecting MS Access ODBC Driver

Make sure the MS Access driver is selected and click on the Next button to continue to the final dialog.



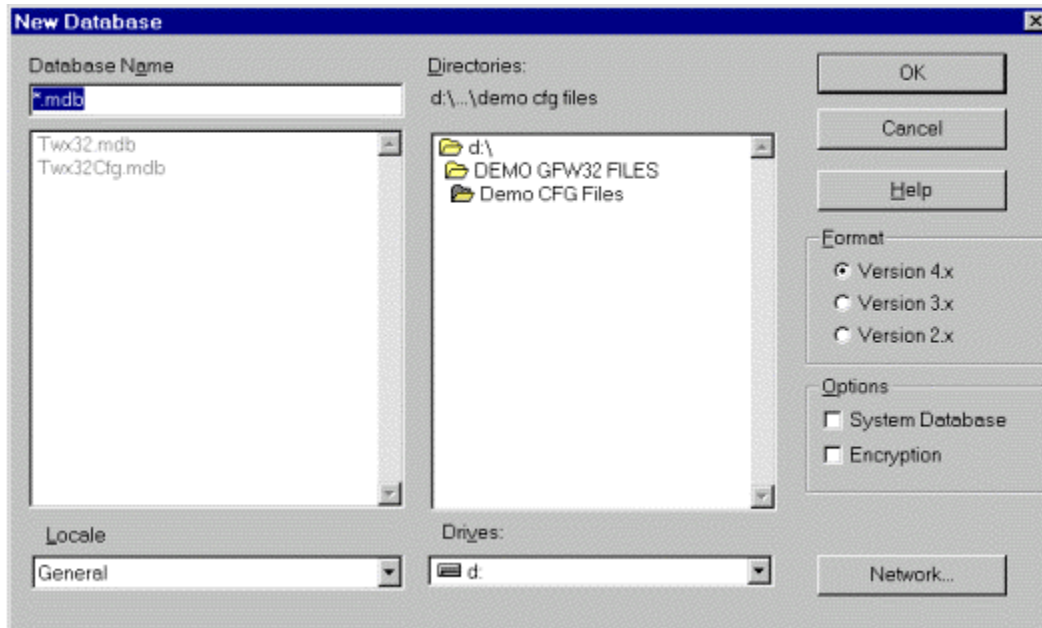
Creating a New MS Access Database Connection Dialog

Click on the Finish button to configure your new MS Access database. The next dialog will prompt you for a database name, description and location for the Access *.mdb file.



Configuring a New MS Access Database Dialog

You can Select an existing *.mdb file and / or Create a new one. If you decide to Create a new one, the mdb file directory selection appears:



Creating a New MS Access mdb file Dialog

You can choose a local hard disk or a networked one as the location of your new *.mdb file. Click on OK to exit the configuration and establish the database connection.

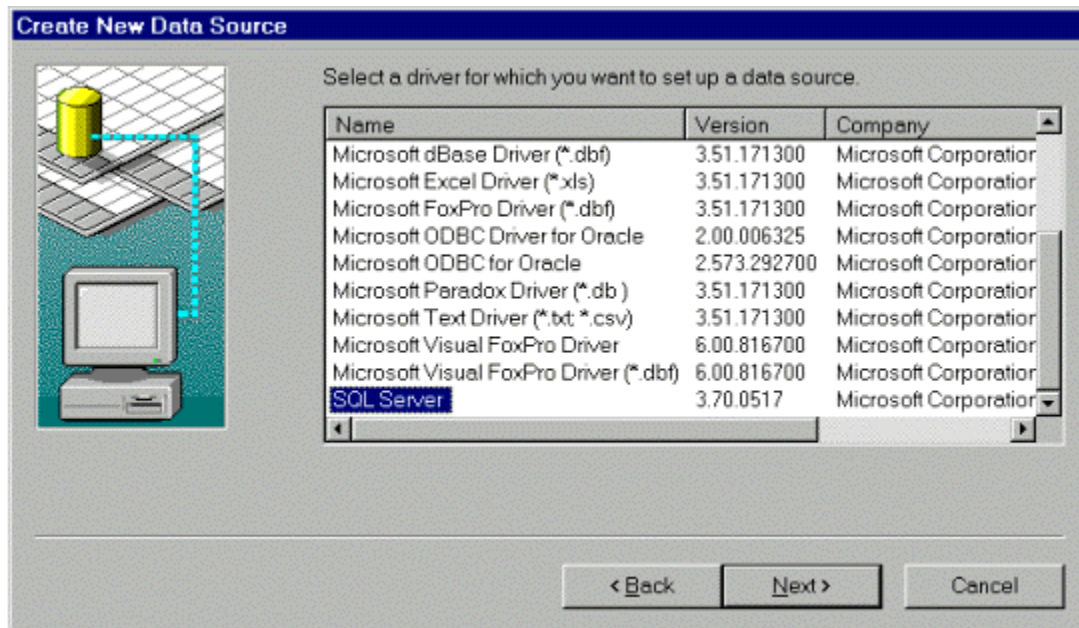
NOTE: When creating a new mdb file, the MS Access configuration dialog presents the options of creating a Version 4.x or a Version 3.x compatible mdb file. By default the Version 4.x is selected, which creates a Jet Engine 4.x compatible mdb file, which can be opened ONLY in MS Access 2000. If the user has MS Access 97, then the Version 3.x compatibility option should be enabled.

Configuring MS SQL Server and MSDE Databases

MSDE (Microsoft data Engine) is a MS SQL Server 7.0 compatible database engine with the following limitations:

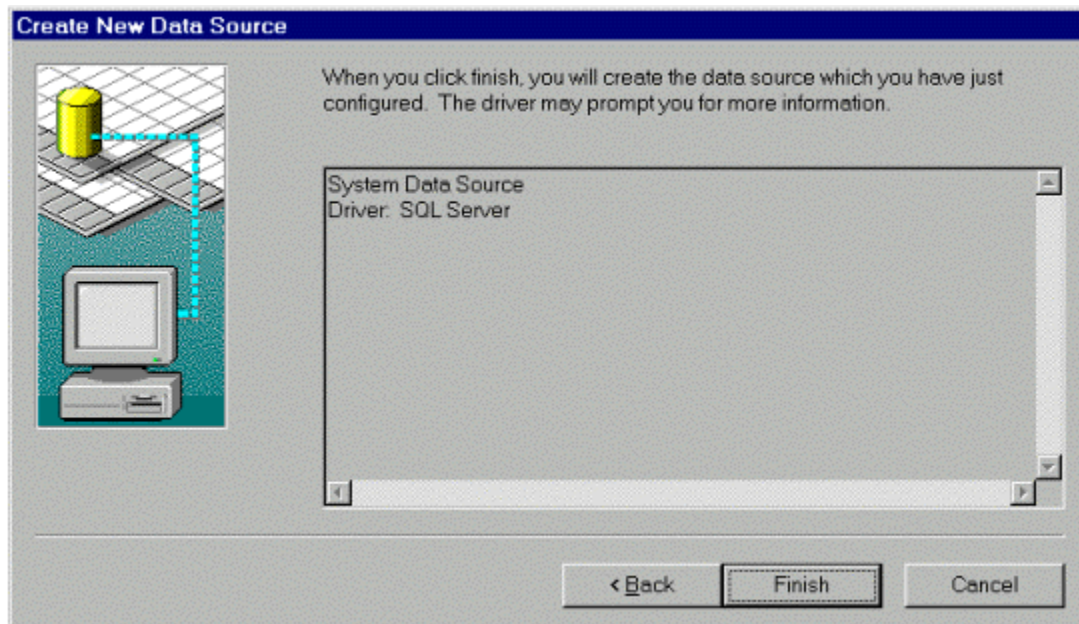
- Maximum database size is 2 GB
- Performance can degrade if more than 5 connections are open at one time

However, MSDE is a fully MS SQL Server 7.0 compatible engine which can be directly upsized to the full MS SQL Server 7.0. As such, database connections to MSDE are identical to those of MS SQL Server 7.0. We assume that by using the Enterprise management tools that come with MS SQL server or the MSDE installation tools available in the ICONICS Product CD there is already a database created on the database Server PC. To establish an ODBC connection to it, when configuring a new ODBC data source for MS SQL Server or MSDE, select the appropriate ODBC driver.



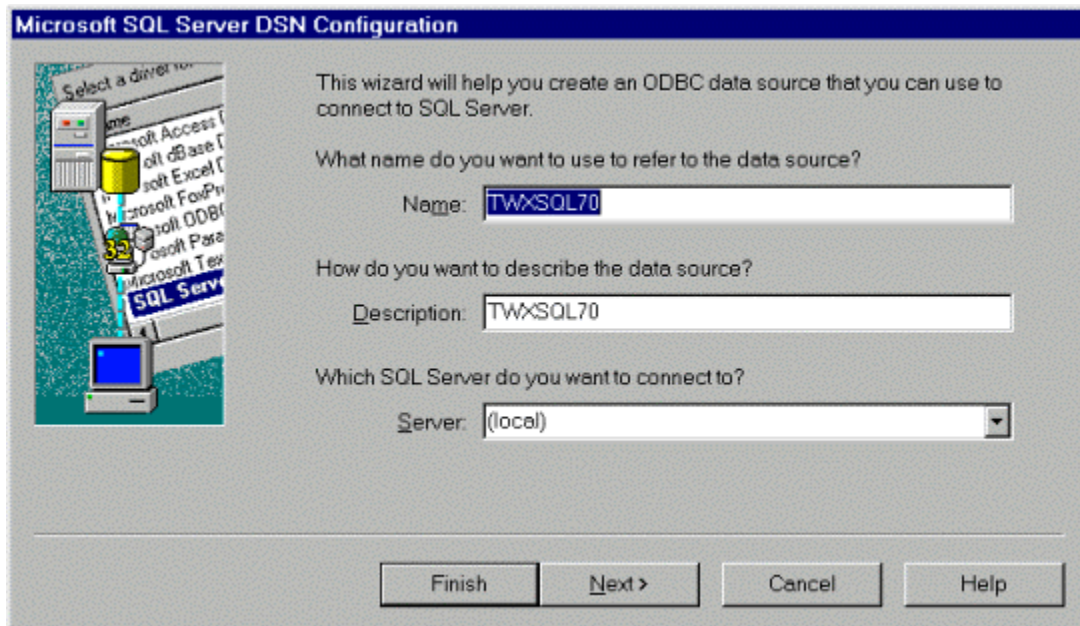
Selecting MS SQL Server ODBC Driver Dialog

Click on Next to continue.



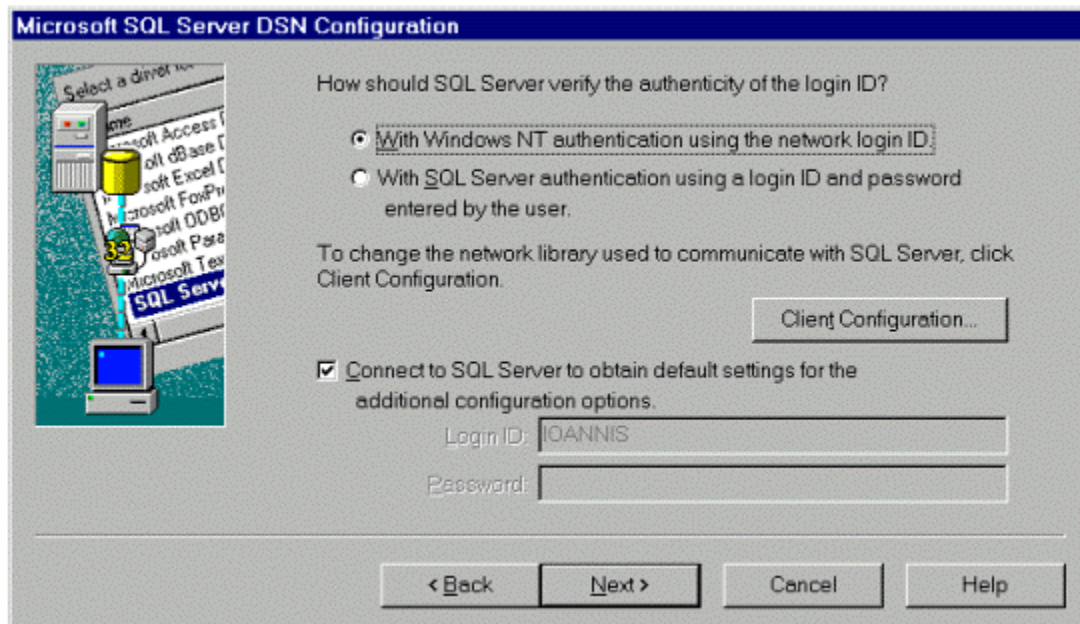
Creating an MS SQL Server / MSDE Database Connection Dialog

Clicking on the Finish button prompts you to continue with the configuration by providing a data source name, description and Server name where the MS SQL Server or MSDE engine resides.



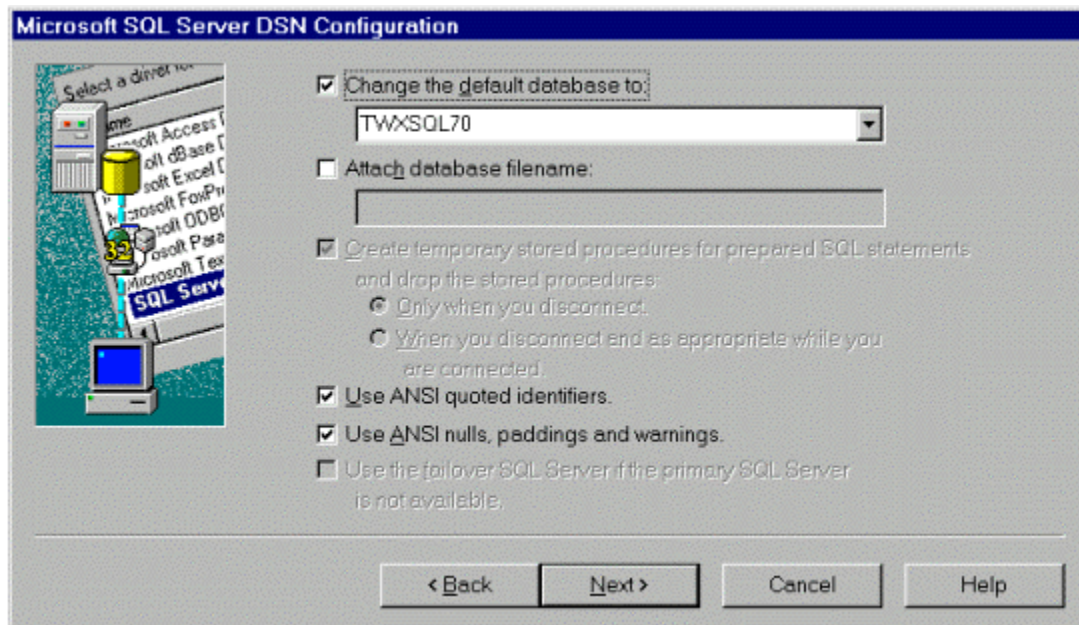
Configuring an MS SQL Server / MSDE Database Connection Dialog

Always select (or input) the appropriate node name where the database resides. Use local only if the database will never be accessed from other nodes.. Click on Next to continue with the MS SQL Server Security configuration and Connection type. We recommend using the Windows NT authentication mechanism, since this eliminates the need for maintaining individual user accounts and passwords.



Configuring an MS SQL Server / MSDE Database Connection Dialog (cont.)

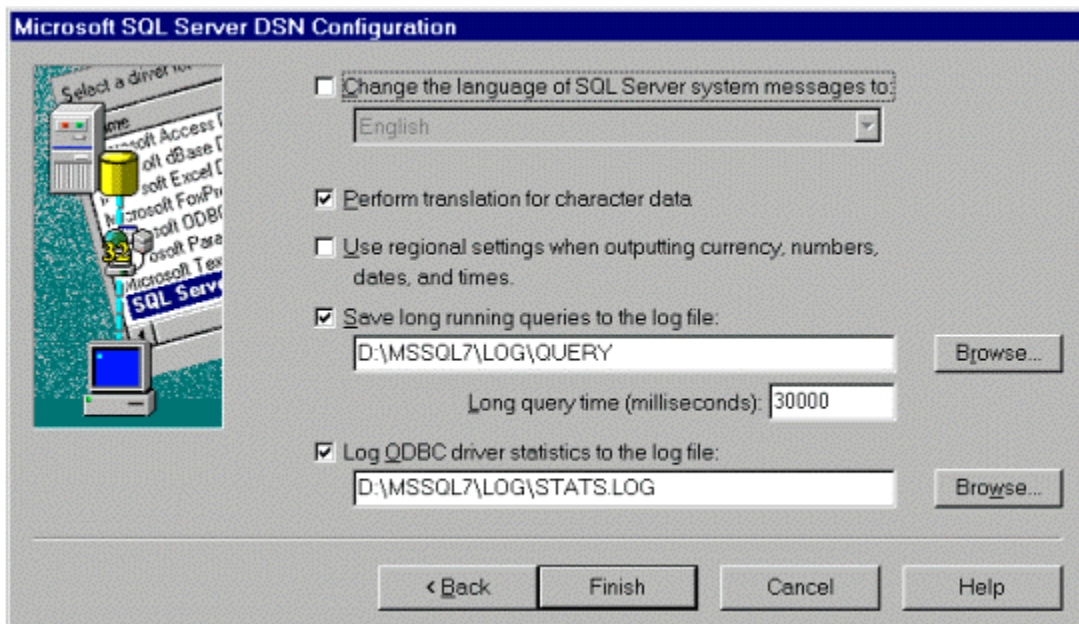
Click on the Next button to continue with the selection of the default database for the connection. Make sure it is not the master database, but the one created for the purpose of alarm logging.



Configuring an MS SQL Server / MSDE Database Connection Dialog (cont.)

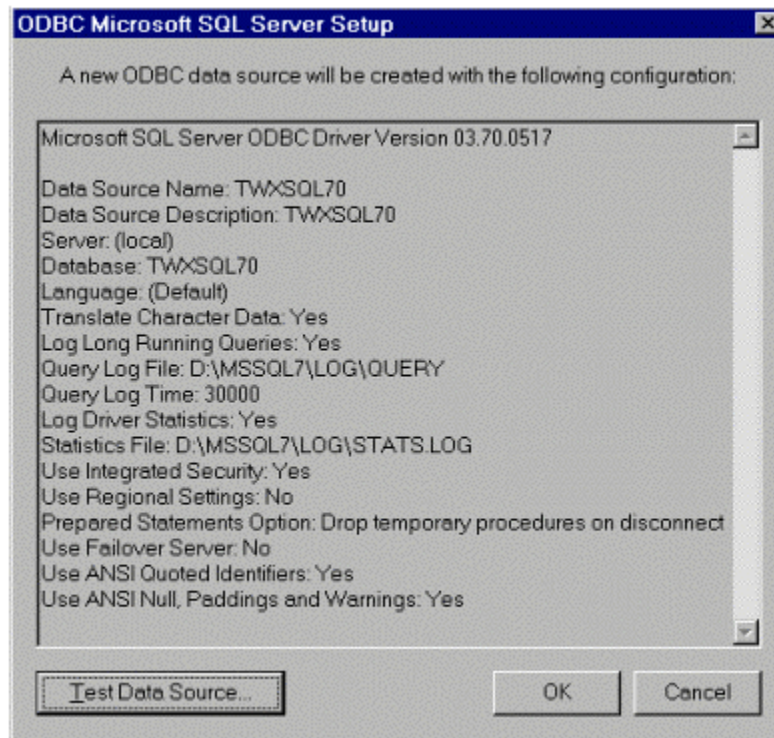
NOTE: When connecting to MS SQL Server or MSDE, make sure that the desired database for alarm logging is selected in the Change the Default Database entry. Otherwise, the user will establish a connection to the master database which is not recommended.

Click on Next to accept or modify the default settings.



Configuring an MS SQL Server / MSDE Database Connection Dialog (cont.)

Finish the connection configuration by clicking the Finish button. You will be prompted to Test the established connection, which we recommend doing.



Testing an MS SQL Server / MSDE Database Connection Dialog

Note: Regardless of the database type, all ICONICS AlarmWorX32 modules, which interact with databases, will create tables, modify tables, drop tables and update tables. Therefore, it is critical that the user currently logged in has all the required rights to perform these tasks.

User Name

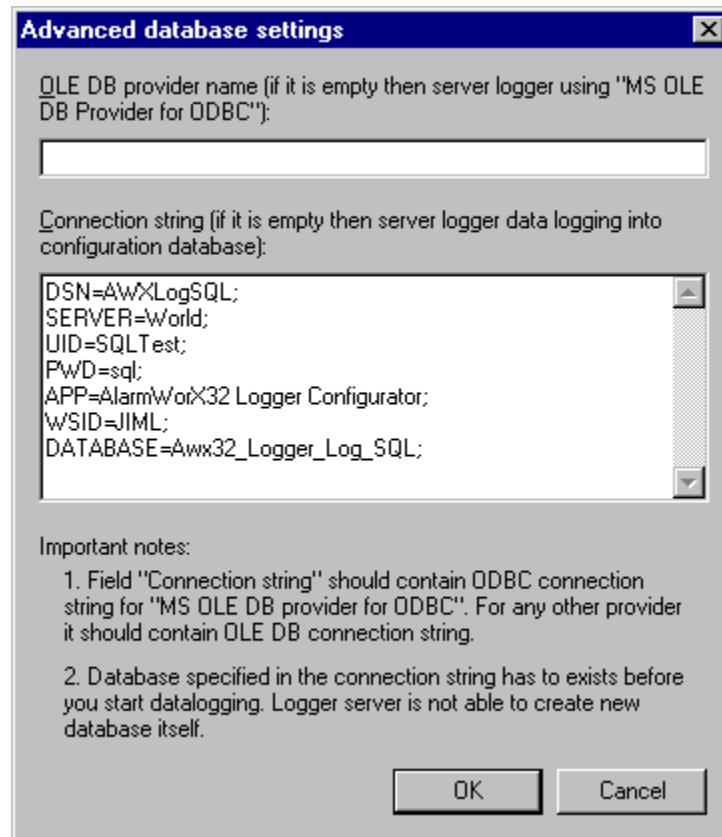
Enter the user name to use (if any) when connecting to the logging database.

Password

Enter the password that goes with the **User Name**.

Advanced Button

The Advanced database settings dialog allows the **OLE DB provider name** and the **Connection string** to be edited directly. This capability is provided to allow connection to databases directly via OLE DB (without using ODBC).



The image shows a Windows-style dialog box titled "Advanced database settings". It has a blue title bar with a close button (X) in the top right corner. The dialog is divided into three main sections. The first section is labeled "OLE DB provider name (if it is empty then server logger using 'MS OLE DB Provider for ODBC'):" and contains a single-line text input field. The second section is labeled "Connection string (if it is empty then server logger data logging into configuration database):" and contains a multi-line text area with a vertical scrollbar. The text in the text area is: "DSN=AwxLogSQL; SERVER=World; UID=SQLTest; PwD=sql; APP=AlarmWorldX32 Logger Configurator; WSID=JIML; DATABASE=Awx32_Logger_Log_SQL;". The third section is labeled "Important notes:" and contains two numbered list items. At the bottom right of the dialog are two buttons: "OK" and "Cancel".

Advanced database settings

OLE DB provider name (if it is empty then server logger using "MS OLE DB Provider for ODBC"):

Connection string (if it is empty then server logger data logging into configuration database):

DSN=AwxLogSQL;
SERVER=World;
UID=SQLTest;
PwD=sql;
APP=AlarmWorldX32 Logger Configurator;
WSID=JIML;
DATABASE=Awx32_Logger_Log_SQL;

Important notes:

1. Field "Connection string" should contain ODBC connection string for "MS OLE DB provider for ODBC". For any other provider it should contain OLE DB connection string.
2. Database specified in the connection string has to exist before you start datalogging. Logger server is not able to create new database itself.

OK Cancel

Advanced Database Settings Dialog

Table Management

General Information on Tables

Data is logged to one master Active table. If no restrictions are placed on the size of the active table, then logging will continue to go to the active table as long as the logger is running. If, however, a restriction is placed on the size of the active table, then all logged data beyond the active table size will get logged to archive tables. The archive tables also have a maximum size determined by a number of records or time interval. When one archive table is full, archiving goes on to the next table. When all archive tables are full, the oldest archive table is deleted and a new one is set up. The name of each archive table is the name of the active table followed by the date and time of the first alarm in the table. In this system, the Active Table will always have the most recent alarm data. A view of the Table Management configuration is shown below.

Table Management

Table Name

Enter the name of the active table to log data to.

Active Table

Here you can specify a limit on the active table size. The limit can be by number of records, or by time interval. If both items are checked, then the table is full when the first of those two conditions is met. The 'Interval' refers to the time difference between the first item logged (not the logging start time) and the most recent item logged. If neither item is checked off, then no archiving will take place.

Archive Tables

In this section, you can enter in the quantity of archive tables and the condition(s) upon which each archive table is full.

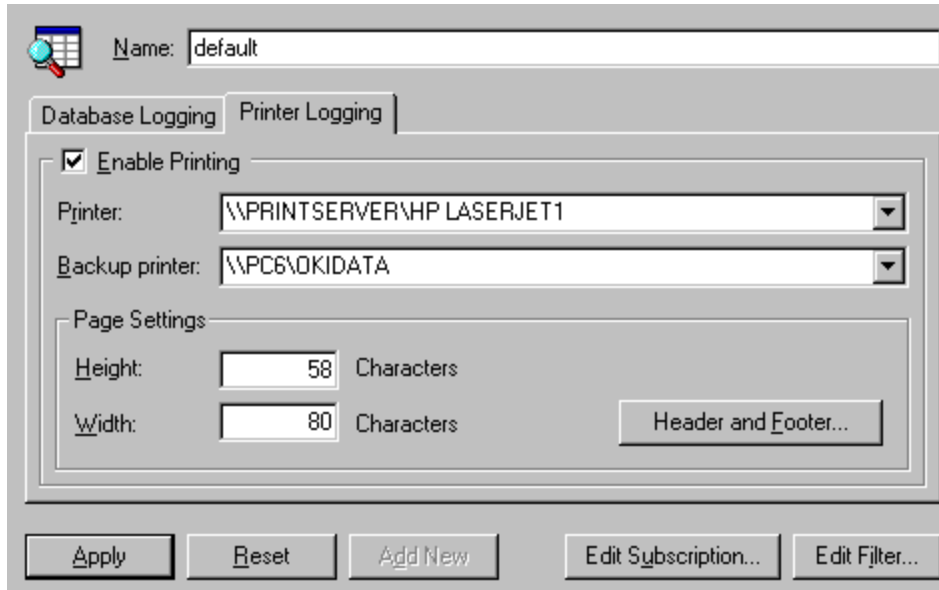
Note: If, in the 'Archive Table(s)' section, the 'Max Records' and 'Max Interval' fields are both left un-checked, then all archiving will go to one archive table of theoretically infinite length as no restrictions have

been placed on the table size. In essence, the 'Max Tables' field will be ignored.

Note: At runtime the logger will create the active table if it does not exist and will also add any columns to the table if they are missing. The logger will not create any indexes. Use a database administration tool to add indexes to columns that are searched and/or sorted by queries.

Note to Microsoft Access Users: If you are connected to a Microsoft Access data base and you are using archive tables, it is very important to note that, though the archiving deletes the oldest table to make room for a new one when all tables are used up, the disk space taken by the deleted table does not get freed up. The end result is that over a long period of time, the log database file will keep growing in size even though you may have specified a finite quantity of records to archive. One solution to this problem is to periodically open the database in Microsoft Access, and use the 'Compact' command. Executing 'Compact' will free up un-used disk space. **It is very important, however, that you make sure that no other users, including the AlarmWorX32 Alarm Logger, are connected to your database when you use the 'Compact' command.** An alternative solution is to use SQL or MSDE for your data logging instead of Microsoft Access. MSDE is included with the Genesis-32 installation.

Printer Logging



Printer Logging Tab

Enable Printing Check Box

Checking this box will cause the Logger to type event records to the printer(s) specified in the following fields.

Printer

Use the drop-down list to select the Primary Printer.

Note: The AlarmWorX32 Logger sends each event directly to the printer as soon as it is received. This works best with a dedicated ink jet or dot matrix printer that will print one line at a time, rather than a page oriented printer like a laser printer. Laser printers will still work, but the page will not be ejected until it is full.

All event notifications will print on the Primary Printer unless an error is detected (offline, out of paper, etc.) in which case the Backup Printer will be used. Switchover from Backup to Primary occurs automatically when the error condition is cleared on the Primary Printer.

Backup Printer

Use the drop-down list to select the Backup Printer.

The Backup Printer will be used whenever an error is detected (offline, out of paper, etc.) on the Primary Printer. Switchover from Backup to Primary occurs automatically when the error condition is cleared on the Primary Printer.

Height

Enter the number of lines your printer prints on a single page using its default font.

The logger prints events "one line at a time", by sending raw text to the printer, bypassing the normal windows page device context printing. Printers that accept this raw text will use an internal font to render the characters. The number of lines that will fit on a page using this internal font needs to be entered here. If you can not find this information in the printer manual, you may need to experiment to find an acceptable value. A value of 58 will be safe for most printers.

Width

Enter the number of characters the printer is capable of printing on a single line using its default font.

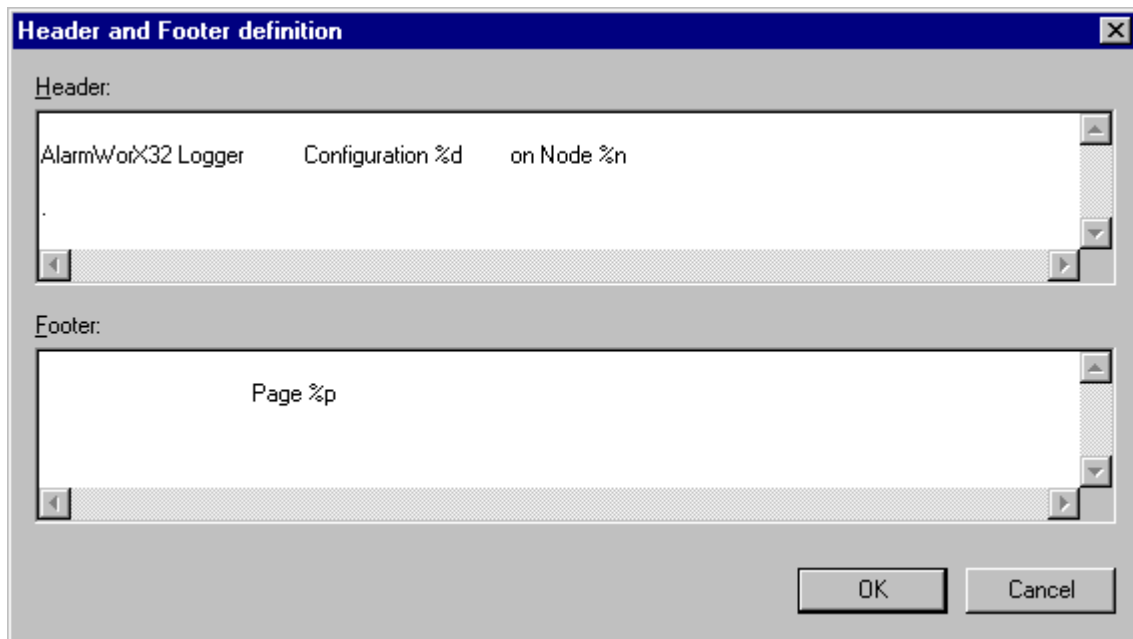
Header and Footer Button

Select the Header and Footer button to edit the lines of text that will appear at the top and bottom of each printed page. The header and footer may contain more than one line. The following tokens will be replaced at runtime with their appropriate values:

%d - Name of the configuration.

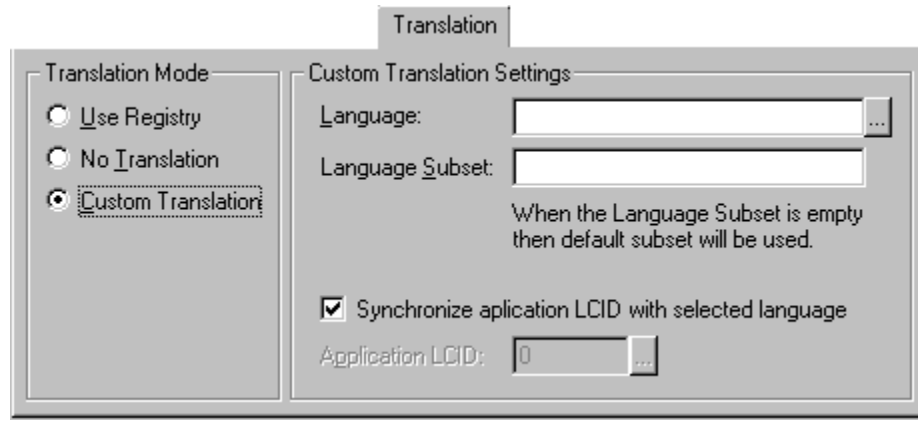
%n - Computer name the logger is running on.

%p - Page number.



Header and Footer Dialog

Translation Tab



Translation Tab

The Translation tab will appear only if you have the version of AlarmWorX32 with Language Switching installed. It is for use with the language translations as configured in the Language Configurator.

Translation Mode allows you to choose how you would like Language Switching to work with the Alarm Logger for this particular configuration.

Use Registry

This mode will use the registry settings to determine what language to use. This means that if a user changes the language in another application, the Alarm Logger will also use the new language.

No Translation

This mode will not allow for the language to be translated before getting logged. If you have items configured with special characters for use with Language Switching, they will remain with the items and be logged in this way. This is useful if you would like the next phase, such as reporting, to be able to determine which language translation to use.

Custom Translation

This mode allows you to set the language translation here so that it will always be the language used with the Alarm Logger for this particular configuration. The following Custom Translation Settings are available for use with this mode:

Language - Choose the language you wish to use from the list of available languages as configured in the Language Configurator.

Language Subset - If you are using a Language Subset, enter it here. It must be a Language Subset that has been defined in the Language Configurator.

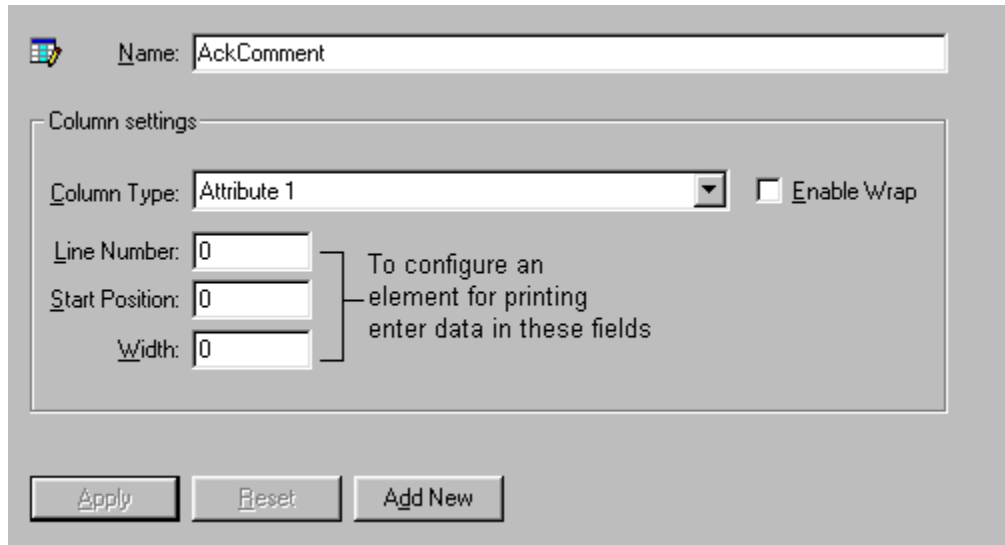
Synchronize application LCID with selected language - If using application language converting also, you can synchronize with language switching by checking this box.

Application LCID- Allows you to choose a language to use for application language converting. This option is only available for use with languages that have alternate language resources (dynamic link libraries) associated with them. See document 'International ToolworX.doc' on your ICONICS CD for more information. It can be found in the CD directory in the Tools folder under Translations.

Columns

Each configuration contains one or more columns. The columns defined for a configuration appear as child elements of the configuration in the tree control, and as list items in the list view (top right pane).

To insert a new column, select the desired configuration in the tree control and select **Edit - New - Column Definition** from the main menu.



Column Configuration Form

The following fields are specified for each column.

Name

Enter the name of the column. This will be the column name used in the logging database and the column heading when printing.

Column Type

Use the drop down to select the OPC Alarm & Events data item you wish to log in this column.

Line Number

When printing, each event may occupy more than one printed line. Specify which line you wish this column to print on. A value of zero will prevent this column from printing.

Start Position

Enter the number of character spaces from the left to begin printing. The first character space is zero.

Width

Enter the number of character spaces for this column when printing.

Enable Wrap Check Box

When printing, field data that is wider than the configured column width will either be truncated or wrapped to the next line depending on this setting.

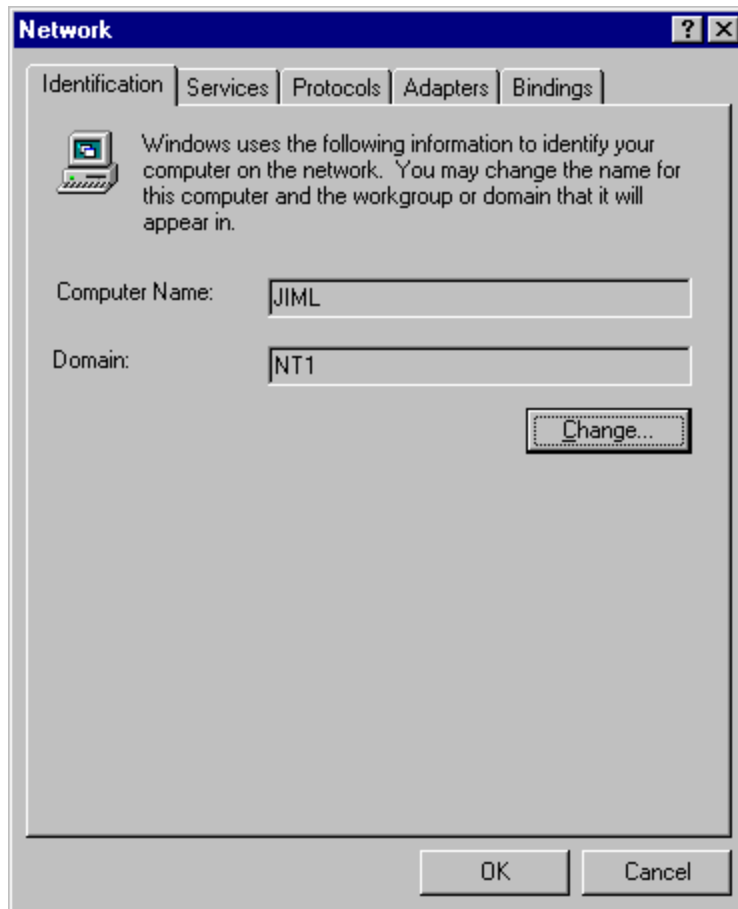
Nodes

Each node represents a computer on the network that is configured to run one or more of the named logger configurations in this database. When the logger executable is run it will look for the its network computer name in the list of Nodes to determine which configuration(s) to run. If a logger can not find its computer name in the list of nodes, it will run the configuration named "default".

To insert a new node, select the "Nodes" entry in the root of the tree control and select **Edit - New - Node** from the main menu.

Name (Node)

Enter the name for the node. The name must match the Computer Name defined in the network configuration. To verify the computer name, select **Network** from **Control Panel** and view the computer name.



Verifying the Computer Name

Assigned Configurations

Each node has one or more configurations assigned to it. The configurations assigned for a node appear as child elements of the node in the tree control, and as list items in the list view (top right pane). A configuration may be assigned to more than one node.

To assign a configuration to a node, select the desired node in the tree control and select **Edit - New - Configuration** from the main menu.

Name (Configuration)

From the drop-down box, select an existing configuration to assign to this node.

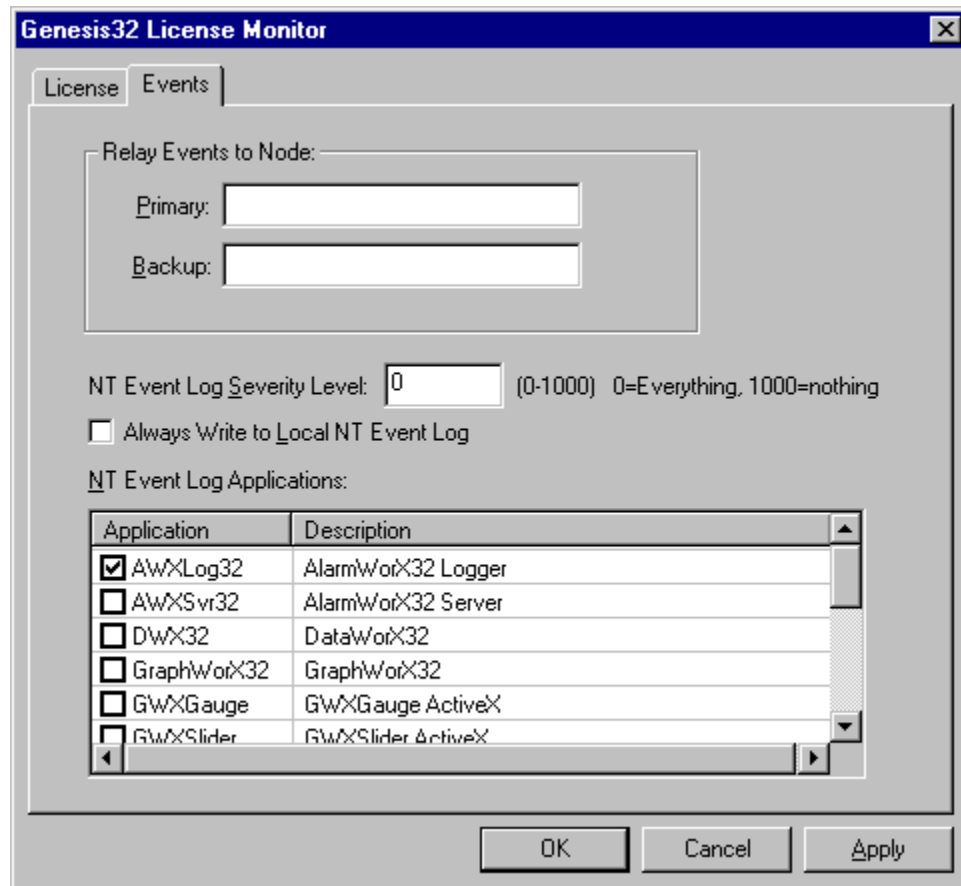
Running the Logger

The Logger Application (AWXLog32.exe) that provides the runtime storage and printing has no user interface and may optionally be run as a service on Windows NT (See the installation section). The logger reads its configuration information from a Microsoft Access Database file. The specific database file the logger will read its configuration from is set by loading the desired database in the Configurator and selected **File - Make Active** from the main menu.

Possible Runtime Errors

Any error conditions encountered by the logger will be reported via the ICONICS Event Server and optionally to the Windows NT Event Log.

To allow the logger to send messages to the NT Event Log, select "License Monitor" from the **GENESIS-32 - Tools** program group, and select the Events tab. Check **AWXLog32** in the list of application.



NT Event Log Configuration

Here are the possible messages:

SEVERITY	SOURCE	MESSAGE
500	AWXLog32	ICONICS AlarmWorX Logger Started.
500	AWXLog32	ICONICS AlarmWorX Logger Stopped.
500	AWXLog32	Primary Printer "%1" now in service. Printing on Backup Printer "%2" terminated
500	AWXLog32	Printing has resumed
500	AWXLog32	Configuration "%1" INSERT INTO table "%2" succeeded.
750	AWXLog32	Failed to Open Primary Printer "%1"
750	AWXLog32	Configuration "%1" printer event queue

SEVERITY	SOURCE	MESSAGE
		overflow. Messages have been lost. Queue size is %2 events.
750	AWXLog32	Primary Printer "%1" has failed. Now Printing on Backup Printer "%2"
750	AWXLog32	Configuration "%1" filter is inactive due to syntax error(s).
750	AWXLog32	Configuration "%1" logging event queue overflow. Messages have been lost. Queue size is %2 events.
800	AWXLog32	Error opening Configuration Database: Provider: %1 Connection: %2 (Error string from database is in "Comment" Attribute)
800	AWXLog32	Error Querying Configuration Database: Provider: %1 Connection: %2 Query: %3 (COM error string is in "Comment" Attribute)
800	AWXLog32	Configuration Database has no configuration for "Node" and no "Default". Provider: %1 Connection: %2
800	AWXLog32	Error opening Logging Database for Configuration "%1" Provider: %2 Connection: %3 (COM error string is in "Comment" Attribute)
800	AWXLog32	Query of Column definitions failed for configuration "%1" Query: %2 (COM error string is in "Comment" Attribute)
800	AWXLog32	Error opening Configuration Database: Provider: %1 Connection: %2 (COM error string is in "Comment" Attribute)
800	AWXLog32	Configuration Database has no configuration "%1". Provider: %2 Connection: %3 (COM error string is in "Comment" Attribute)
800	AWXLog32	Configuration "%1" subscription to OPC Event Server(s) failed.
800	AWXLog32	Configuration "%1" error assigning value to column "%2"
800	AWXLog32	Configuration "%1" INSERT INTO table "%2" failed (Error string from database is in "Comment" Attribute)

SEVERITY	SOURCE	MESSAGE
800	AWXLog32	Failed to register "%1" with GenRegistrar. Configuration "%2" will not execute.
800	AWXLog32	Printing to both Primary Printer "%1" and Backup Printer "%2" has failed



Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.

Foxborough, MA 02035

Tel: 508-543-8600

FAX: 508-543-1503

E-MAIL: support@iconics.com

WEB: <http://www.iconics.com>

Contents

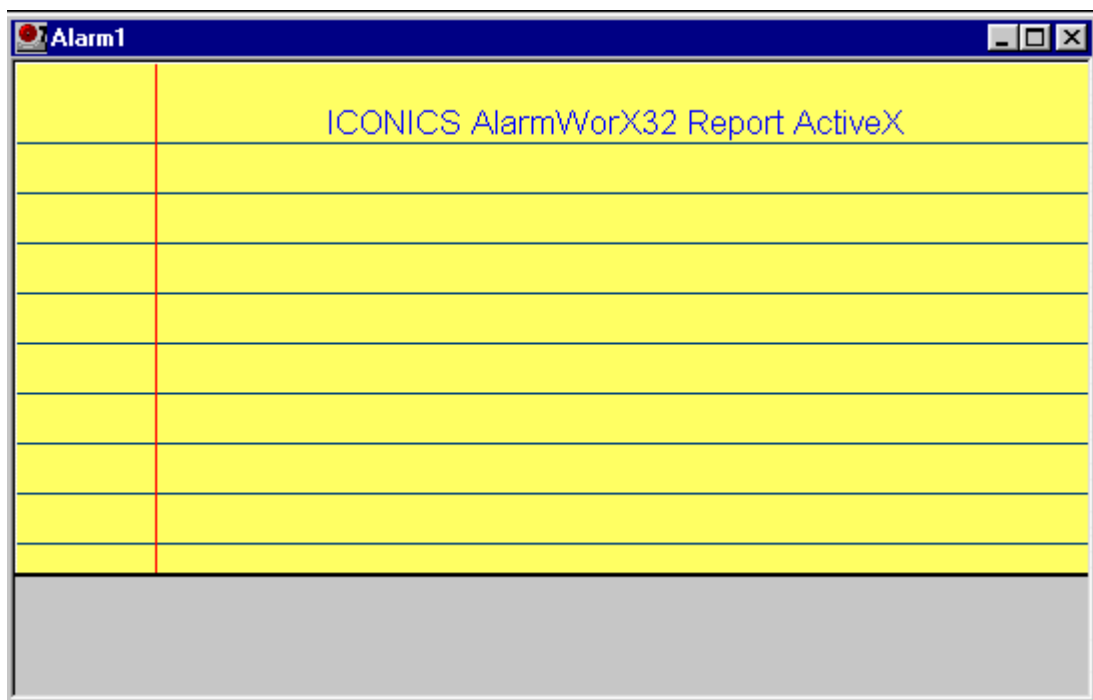
AlarmWorX32 Report

Runtime Mode

AlarmWorX32 Report

The AlarmWorX32 Report allows reporting (user configured or pre-configured) and graphing of alarms. The source of the Alarm data is alarms previously logged by the Alarm Logger. The Alarm Report is an ActiveX, allowing it to be placed in any ActiveX container application, including GraphWorX32, VB or a web page.

The AlarmWorX32 Report ActiveX is accessible from the ActiveX toolbar in the AlarmWorX32 Container, or from the **Edit - Insert New Object** menu item. If you select the latter the 'Insert New Object' dialog will appear. Select the ICONICS AWXRep32ActiveX item. Once the Report ActiveX has been inserted into the display the window should look like the following figure.



Alarm Container Window With the AlarmWorX32 Report ActiveX

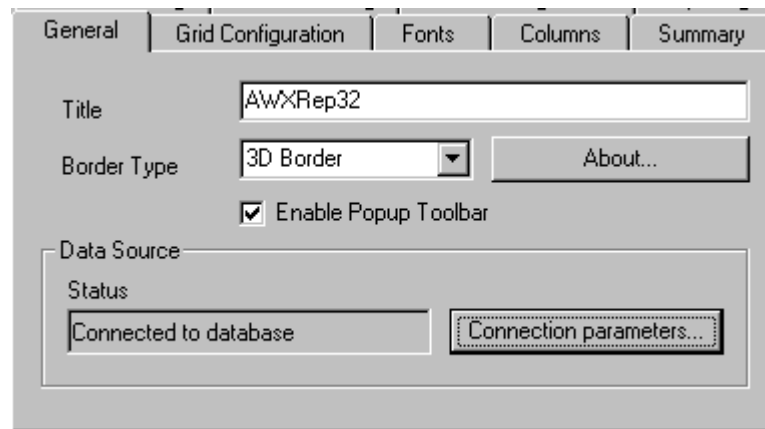
Configuration

To start the configuration of the Report ActiveX double click on the inserted item (the yellow notebook type object) and the Property Pages for this ActiveX will appear. Using these property pages the user will configure how alarms will be reported (appearance, organization, etc.) and which report type will be used.

General
Columns
Grid Properties
Charts Configuration
Fonts
Records Sorting
Record Filtering
Reporting

General

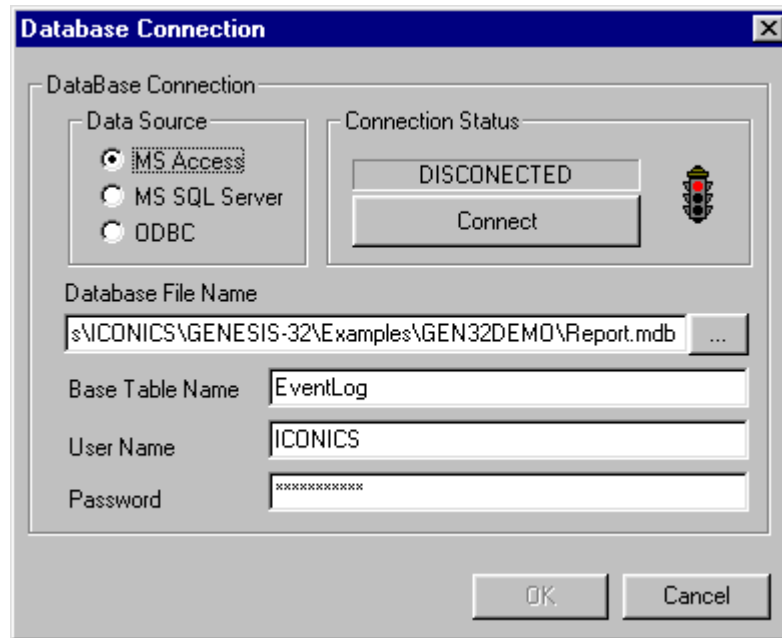
The General tab is where the user is able to enter the name of the report file, select the border type and indicate the data source.

The image shows a screenshot of the 'Report ActiveX General Tab' configuration window. The window has a tabbed interface with five tabs: 'General', 'Grid Configuration', 'Fonts', 'Columns', and 'Summary'. The 'General' tab is currently selected. Inside the 'General' tab, there is a 'Title' text box containing 'AwXRep32'. Below it is a 'Border Type' dropdown menu set to '3D Border', with an 'About...' button to its right. A checkbox labeled 'Enable Popup Toolbar' is checked. The 'Data Source' section contains a 'Status' text box showing 'Connected to database' and a 'Connection parameters...' button.

Report ActiveX General Tab

Clicking the 'Connection parameters...' button will bring up the **Database Connection** dialog as shown below. The user is then able to select from MS Access, MS SQL Server or ODBC (Open Data Base Connectivity) in the 'Data Source' field. If MS Access is selected, you are able to type in the database file name or click the '...' button to find and open the MS Access

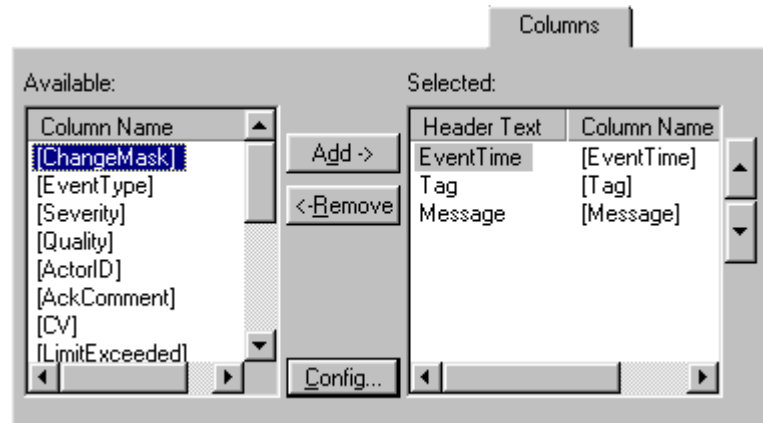
database you wish to use for the report. There are also fields in which to enter a user name and password for your database. You can simply press the 'Connect' button to connect to the specified database. If MS SQL is selected, then pressing the 'Connect' button will launch a dialog box prompting you choose the MS SQL Server you wish to use for the report. If a wrong or corrupt database file is selected a warning box will appear indicating this to the user. If ODBC is selected, then pressing the 'Connect' button will launch a window from which to select an ODBC data source.



DataBase Connection

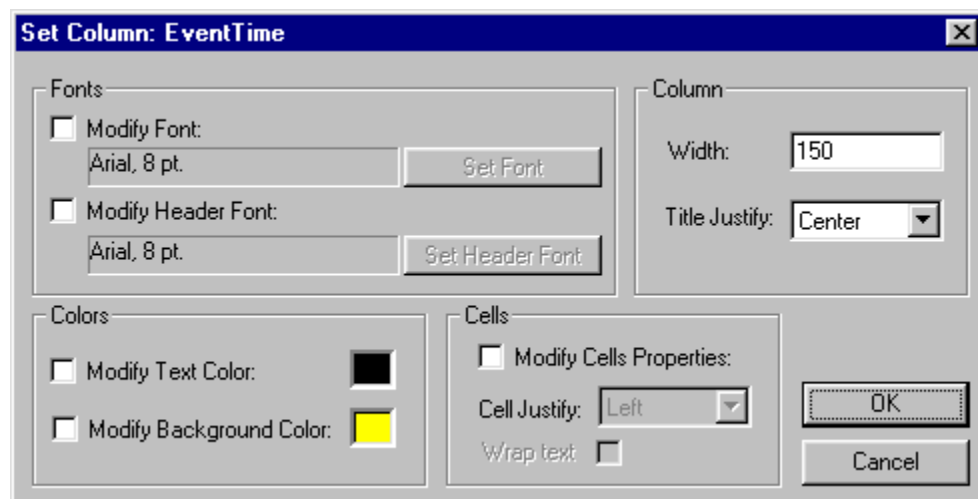
Columns

The Columns tab indicates which columns will appear in the report. By default the following three items are included in the report: EventTime, Tag, and Message.



Report ActiveX Columns Tab

Using the buttons in the middle of and the arrows on the right side of this property page, the user can select which columns they want to include in the report as well as indicating the order of their appearance. To move an item from the 'Available' list to the 'Selected' list, simply press the 'Add' button. To move an item back from the 'Selected' list to the 'Available' list, press the 'Remove' button. To change the order of an item in the 'Selected' list, simply select the item whose order you wish to change and press the up and down arrows located on the right-hand side of the form to change its position. You can modify the appearance of a column by either double-clicking on an item in the 'Selected' list or by selecting an item and pressing the 'Config...' button. A configuration dialog like the one shown below will appear. From this dialog, you can set the font, color, width, and cell properties by selecting and changing the various options available.



Columns Configuration Dialog

Grid Properties

This tab allows the user to configure how the grids on the report will look and act.

The screenshot shows the 'Grid Configuration' dialog box. It is organized into several sections:
- **Appearance:** Includes radio buttons for 'Flat' and '3D'.
- **Row Height:** A numeric input field showing '14'.
- **Colors:** Includes color pickers for 'Text Color', 'Header Color', 'Text Back Color', and 'Grid Back Color'.
- **Title:** Includes a checked checkbox for 'Display Grid Title', and color pickers for 'Title Text Color' and 'Background Color'.
- **Horizontal Grid Lines:** Includes a 'Style' dropdown set to 'Dot', a 'Width' spinner set to '1', and a 'Color' picker.
- **Vertical Grid Lines:** Includes a 'Style' dropdown set to 'Dot', a 'Width' spinner set to '1', and a 'Color' picker.

Report ActiveX Grid Configuration Tab

Charts Configuration

The Charts Configuration tab refers to the chart that will be generated from the report data. The user has two choices for the type of chart: Bar or Pie. Additionally they are allowed to select the background color for the chart (the color that will surround the chart) and the chart's title. The user can toggle visibility of chart title and/or legend by checking or unchecking 'Display Title' or 'Display Legend,' respectively.

Exact positions and sizes of chart elements (title, plot, and legend) can be set. The positions are determined by values in the 'Left' and 'Top' fields, and sizes for particular chart elements are set in the 'Height' and 'Width' fields. The positions and sizes are given in the percentage of full width and height of chart area, assuming that coordinate origin (0,0) is located in the top left corner of the chart. For example setting Left=50, Top=50, Height=50, Width=50 for the chart legend will position legend in the lower right quarter of the chart area.

If the 'Update Title On Series Select' box is checked, the chart's title text will be updated when user clicks on any data series (either bar or pie segment) of the plot object. A name of the selected data series will be included with the title text. This mode is particularly useful when the chart legend is hidden.

Report ActiveX Charts Configuration Tab

In Runtime these features will determine how the chart will appear to the user. To access the Report Chart, enter Runtime and click the Chart Summary button. For more details on this runtime feature please see the **Runtime - Chart Summary** section.

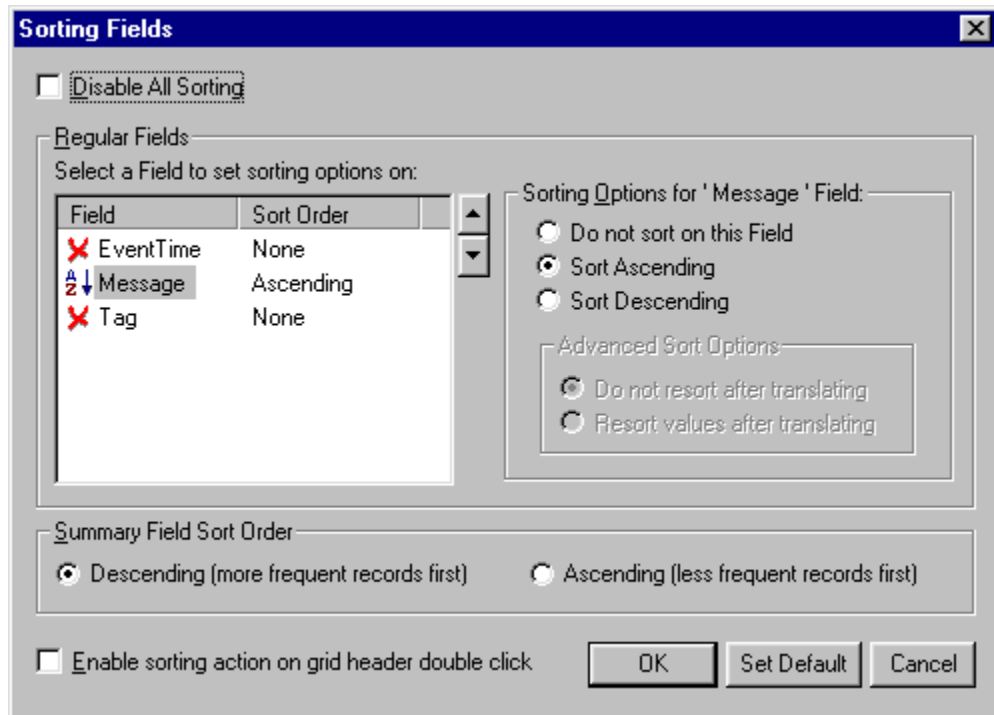
Fonts

The Fonts tab is a typical fonts format screen.

Report ActiveX Fonts Tab

Records Sorting

The Sorting tab has one button - the 'Sort By' button. Click this button and the following dialog appears.



Sorting Fields Dialog

In the sorting dialog, the user has the option of choosing whether or not to disable or enable sorting by leaving the 'Disable All Sorting' checkbox checked or unchecked. If the user chooses to leave this box unchecked, then there are several sorting options available. To change the sorting method of one of the regular fields, simply select the desired field from the list box and pick from the three sorting options to the right (ascending, descending, no sort). To change the order of sorting priority, select an item in the list box and press the up and down arrows located to the right of the list box to move the item up and down the list. All elements will be sorted by the first category in the list. In the event that two elements in the first category are identical, the reported alarms are sorted by the criteria configured for the next item in the list. If you are dealing with language translations, you have the option of resorting values after translating them. This option is for advanced users and is not recommended for scenarios where there is a large set of records after filtering. This dialog also contains the option of setting the sort order for the 'Summary' field (see **Runtime - Data Grid** and **Chart Summary**) for more information on the summary field) Simply select 'Descending' or 'Ascending'. If the 'Enable sorting action on grid header double-click' checkbox is selected, then the user has the option of double-clicking the header of each column in the report while in runtime to toggle between ascending and descending sorting for the selected column.

Note: You can also access the Sorting Options dialog box via the 'Sort Alarms' button on the AWXRep32 Toolbar while in Runtime Mode.

(Double click anywhere in the ActiveX client box to show the AWXRep32 Toolbar.)

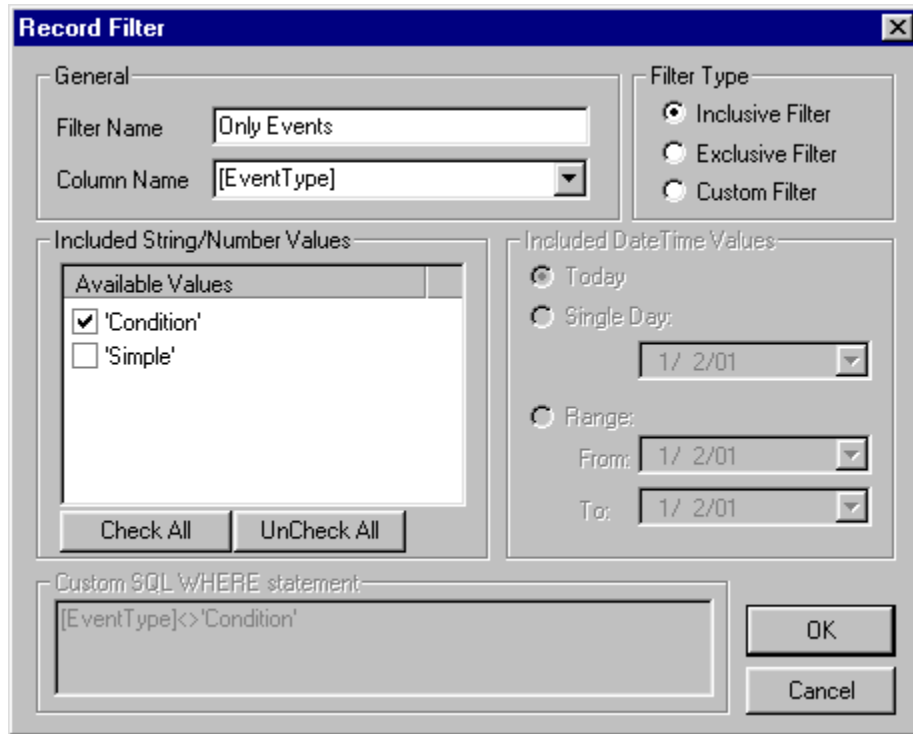
Record Filtering

The Record Filtering tab offers simple filtering options to the user. You can filter by date, by tag or by custom SQL statements.



Alarm Reports ActiveX Filtering Tab

All of the items checked off in the 'Available Filters' list box will be used as criteria for determining which alarms are displayed on the grid in runtime. If the 'Display active filter name(s) on grid's title bar' checkbox is selected, then the names of the filters checked off will appear on the grid's title bar while in runtime. To configure a filter, the user can either press the 'Add Filter...' button to set a new filter or select an existing filter and press the 'Config Filter...' button to modify a filter. Either action will bring up the dialog shown below.



The image shows a 'Record Filter' dialog box with a blue title bar and a close button. It is divided into several sections:

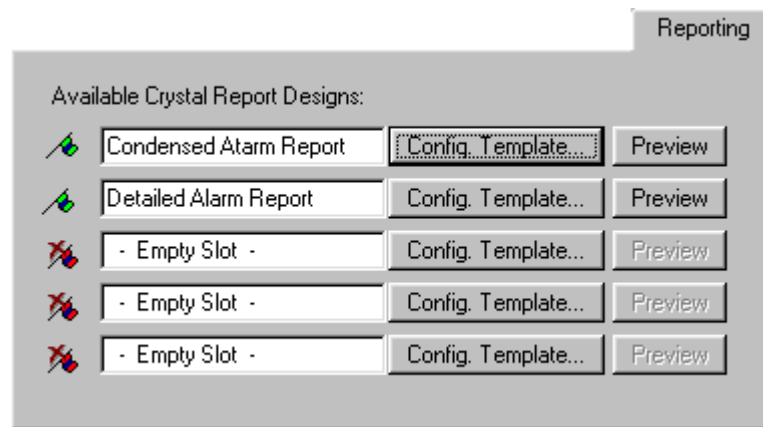
- General:** Contains 'Filter Name' (text box with 'Only Events') and 'Column Name' (drop-down menu with '[EventType]').
- Filter Type:** Contains three radio buttons: 'Inclusive Filter' (selected), 'Exclusive Filter', and 'Custom Filter'.
- Included String/Number Values:** Contains a list box titled 'Available Values' with two items: 'Condition' (checked) and 'Simple' (unchecked). Below the list box are 'Check All' and 'UnCheck All' buttons.
- Included DateTime Values:** Contains three radio buttons: 'Today' (selected), 'Single Day:', and 'Range:'. The 'Single Day:' option has a date picker showing '1/ 2/01'. The 'Range:' option has 'From:' and 'To:' date pickers, both showing '1/ 2/01'.
- Custom SQL WHERE statement:** A text box containing the SQL statement '[EventType]<>'Condition''. Below this text box are 'OK' and 'Cancel' buttons.

Record Filter Dialog

The three types of filtering available to the user are inclusive filtering, exclusive filtering and custom filtering. If an inclusive filter is used, then the alarm conditions that meet the criteria set by the items selected in the list box or the alarms appearing within the time range specified in the 'Included DateTime Values' section pass through the filter and are displayed on the grid in runtime. If an exclusive filter is used, then the matching alarms do not pass through the filter and are not displayed in runtime. The decision of whether to use string/number values or date/time ranges for filtering depends on the column selected in the 'Column Name' drop-down list box. If '[EventTime]' is selected, then a date/time range specified in the right-hand side of the dialog box is used for the filtering criteria. For any other selection, the string/number values checked off in the list box are used for the filtering criteria. You will notice, as you select different values for 'Column name' that the list of available values in the list box changes. If custom filtering is used, then the user has the option of typing an SQL WHERE statement in the text box at the bottom of the dialog to specify what type of values can pass through the filter.

Reporting

The Report ActiveX Control makes use of Seagate Crystal Reports (version 8.0) for versatile, neat and easy to manage alarm reports. From the 'Reporting' tab, a user can configure templates for Crystal Reports. Pressing the 'Preview' button next to any configured template will launch Crystal Reports and display a sample report matching the criteria set in the template configuration.



Alarm Report ActiveX Reporting Tab

Upon clicking on any of the 'Config. Template...' buttons, a dialog will appear with tabs labeled 'Report Fields', 'Title Section', 'Data Section', 'Size', 'Records Filtering' and 'Sorting.' Follow the links below for information on what each of these five tabs does.

Report Fields

Title Section

Data Section

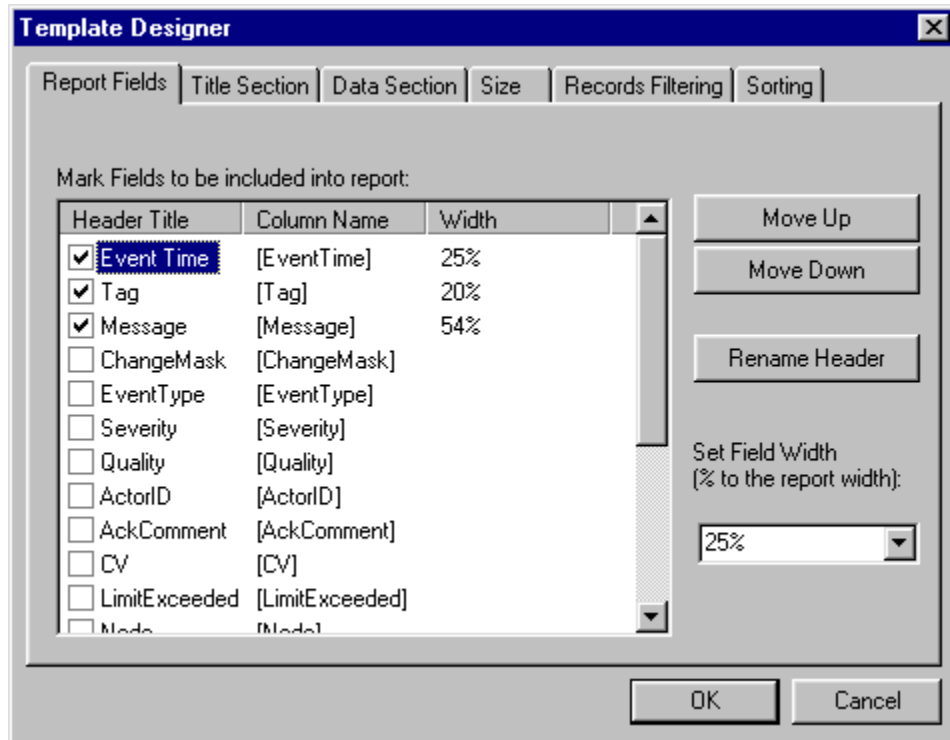
Size

Records Filtering

Sorting

Report Fields

As the 'Template Designer' dialog loads, it defaults to the 'Report Fields' tab. Here you can configure which alarm columns to report, the order in which they are to be reported and the amount of space each category is to take on the report. The 'Report Fields' tab is shown below.

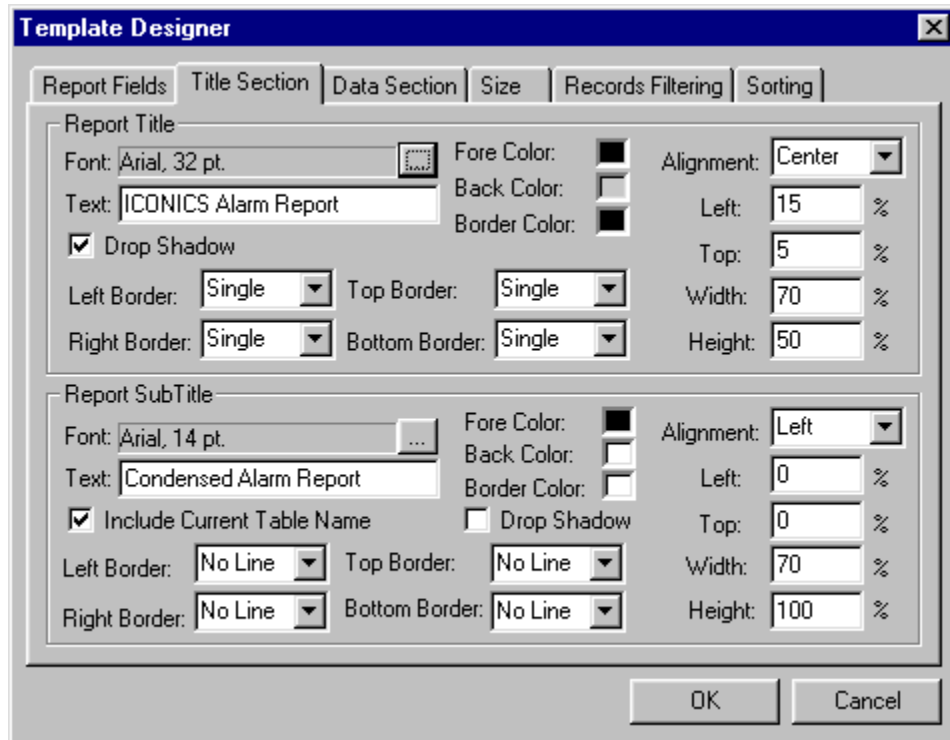


Report Fields Tab of Template Designer Dialog

To select columns to be reported, simply check off the appropriate items in the list box. To change the order of an item, simply select the item whose order you wish to change and press the 'Move Up' and 'Move Down' buttons. You will see the item move up and down the list as you do so. When you select an item, you can also specify a width in terms of a percentage of the width of the entire report. You can re-name the header title by clicking the 'Rename Header' button and filling in the appropriate text or you can simply click on the header text itself and wait for the cursor to appear.

Title Section

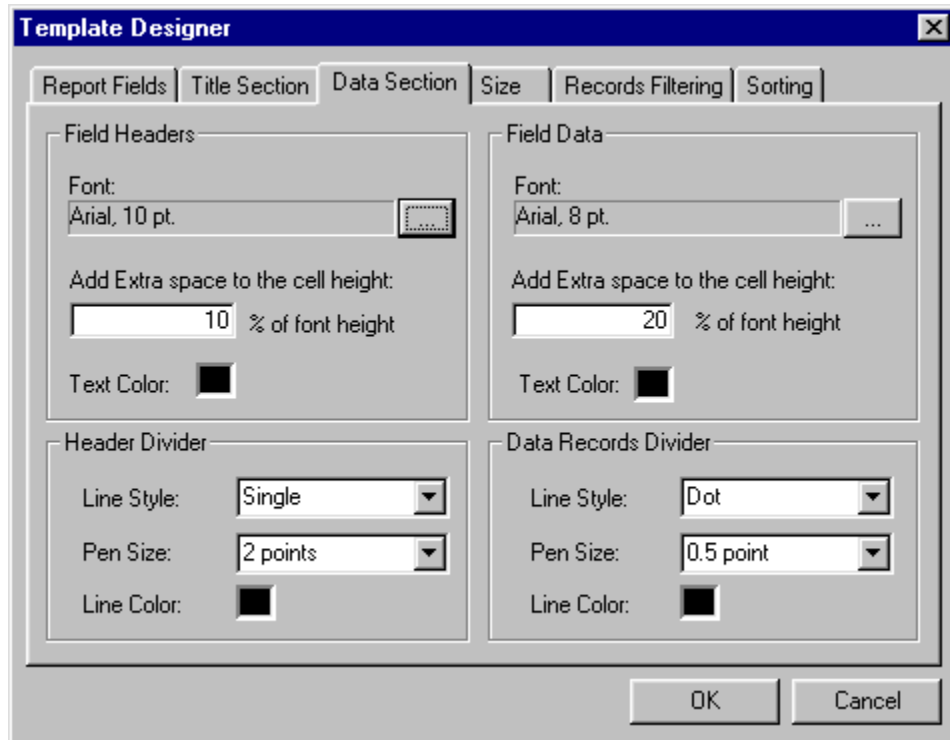
The 'Title Section' tab allows you to configure a title for your report. Simply adjust the text, colors, borders, font, shadowing and alignment of your title and subtitle to match the desired appearance.



Title Section Tab of Template Designer Dialog

Data Section

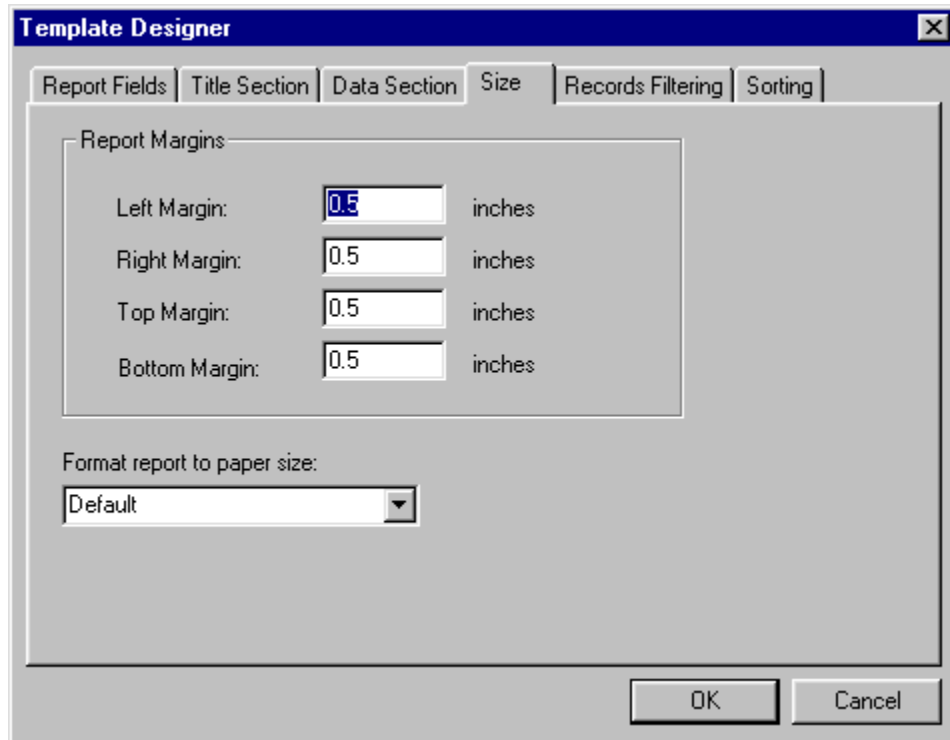
The 'Data Section' tab is where you can configure the appearance of the headers and fields of your reported data. Adjust the font, cell padding and text color of the column headers in the 'Field Headers' frame. You can change the style, thickness and color of the headers dividers by adjusting the 'Line Style', 'Pen Size' and 'Line Color' items in the 'Header Divider' frame. To adjust the appearance of the reported data and dividers, make the appropriate modifications to the 'Field Data' and 'Data Records Divider' frames.



Data Section Tab of Template Designer Dialog

Size

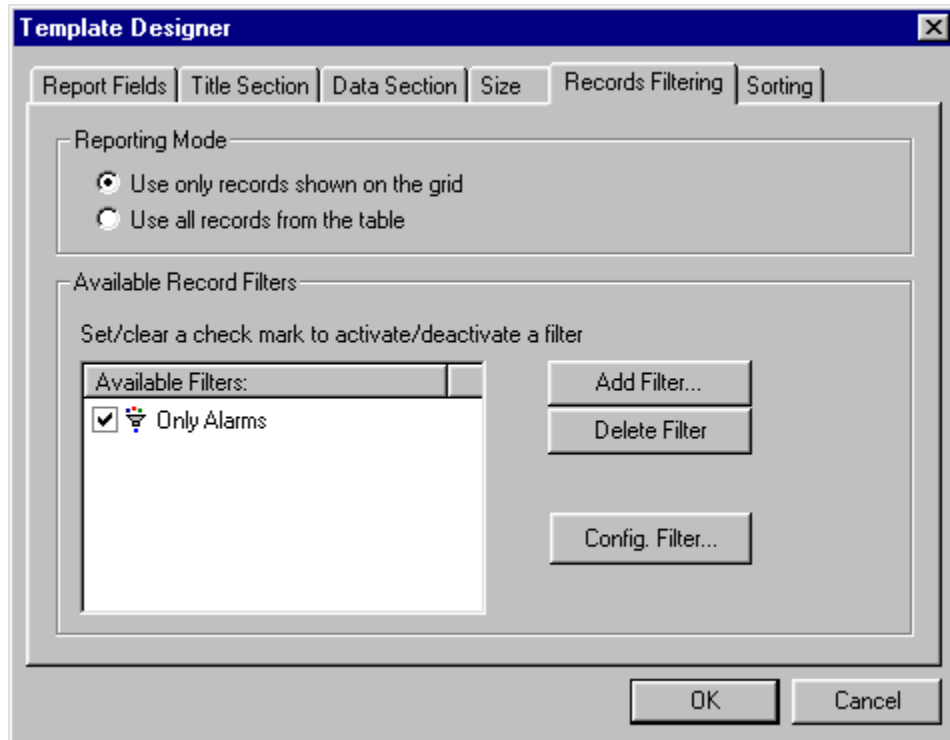
The 'size' tab is where you can adjust the margins for the final report. Simply fill in the margin fields as needed. Adjust paper size by selecting an item from the 'Format report to paper size' drop-down box. (Default is 8.5 x 11 inches)



Size Tab of Template Designer Dialog

Records Filtering

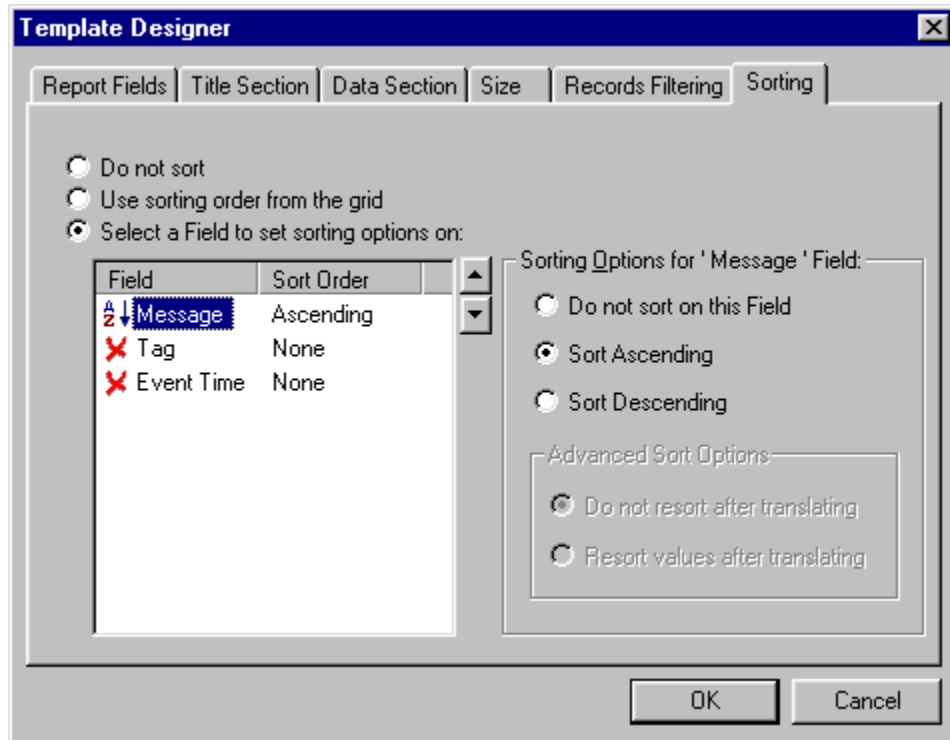
You can configure filters for your reports on the 'Records Filtering' tab. Records can be reported either from the grid or from the alarm table. One level of filtering marks the difference between reporting alarms from the grid and reporting alarms from the table. The alarms that make it from the table to the grid are the ones that pass through the filter defined in the **Record Filtering** section of the ActiveX properties. The records chosen, whether from the alarm table or from the grid, can be filtered for reporting in the 'Available Record Filters' section of the dialog. (In the case where you are reporting only records shown on the grid, this is a second level of filtering from the alarm table to the report) The adding, deleting, and configuring of filters is handled the exact same way as on the **Record Filtering** property tab.



Records Filtering Tab of Template Designer Dialog

Sorting

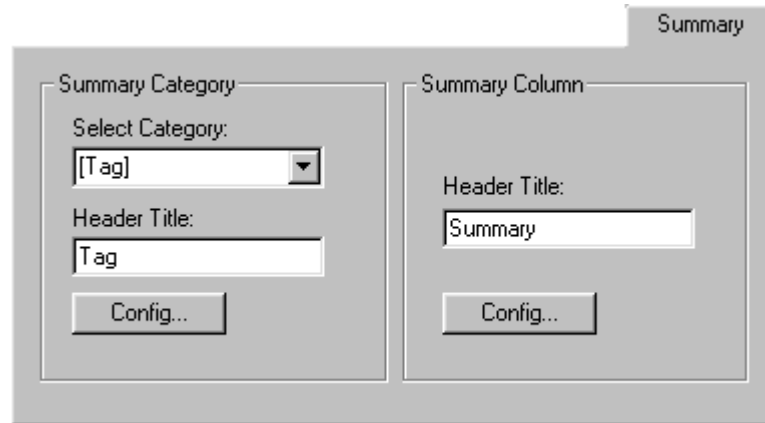
When sorting the records to be reported, there are three options. Users can choose not to sort at all by clicking the 'Do not sort' option. If the 'Use sorting order from the grid' option is selected, then the report sorting defaults to the sorting defined on the **Records Sorting** tab of the Report ActiveX property pages. You can over-ride the grid sorting by selecting the 'Select a Field to set sorting options on' option. Set your sorting parameters with the list box and options below. For detailed information on configuring your custom sorting see the **Records Sorting** section.



Sorting Tab of Template Designer Dialog

Summary

The 'Summary' property page is for the configuration of the content and appearance of summary information to be displayed on the grid in runtime. The 'Select Category' drop down list box is used to select the category of alarm to be displayed in the summary. You can also select a title and press the 'Config...' button on the left to adjust the font, color and appearance of the alarm category selected. The title and configuration on the right are used to modify the appearance of the values associated with the category chosen on the left.



The screenshot shows a configuration window titled 'Summary'. It is divided into two main sections: 'Summary Category' on the left and 'Summary Column' on the right. The 'Summary Category' section contains a 'Select Category:' label above a dropdown menu showing '[Tag]'. Below this is a 'Header Title:' label above a text box containing 'Tag'. At the bottom of this section is a 'Config...' button. The 'Summary Column' section contains a 'Header Title:' label above a text box containing 'Summary'. At the bottom of this section is a 'Config...' button.

Alarm Report ActiveX Summary Tab

Security

To access the ICONICS security server, either select **Tools - Security - Configuration** or click the Security icon on the toolbar. The following dialog will appear.

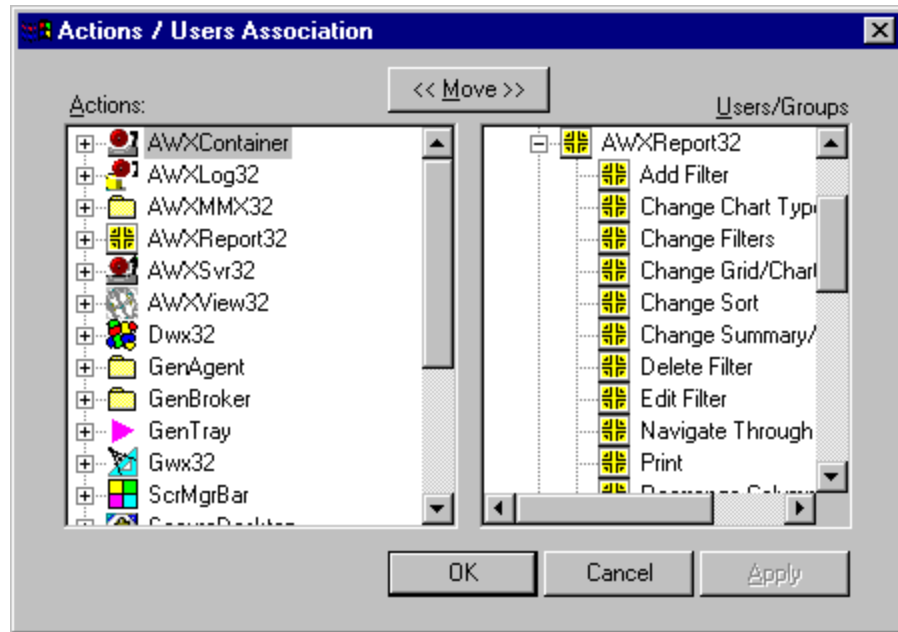


The dialog box is titled "ICONICS Security Server Administrator Login". It contains a padlock icon and the following text: "Enter an Administrator user name and password, or leave the user name blank and enter the default Administrator password:". Below this text are three input fields: "User Name:", "Password:", and "Challenge:". The "Challenge:" field contains the value "18303". At the bottom of the dialog are two buttons: "OK" and "Cancel".

Enter your User Name and Password and click OK. This will bring you to the ICONICS Security Server, where it will be possible to access the AWXReport32 Security Server. Depending on your configured level of security, you may or may not be able to edit components of the AWXReport32 Security Server.

Note: For more information about security, please see the Security chapter of the AlarmWorX32 manual.

Once a user has gained access, the ICONICS Security Server dialog box will appear. Under the Edit menu, choose Application Actions. The Actions/Users Association dialog box will appear as shown below.



The security server can be set up to allow access to several different actions from this dialog. To set the options for the AlarmWorX32 Report select AWXReport32 from the list in the right pane. This will bring up the list seen in the left pane, which includes the following options:

- Add Filter
- Change Chart Type
- Change Filters
- Change Grid/Chart View
- Change Sort
- Change Summary/Details View
- Delete Filter
- Edit Filter
- Navigate Through Tables
- Print
- Rearrange Columns
- Resize Columns
- Run Crystal Reports
- Toggle Toolbar

Runtime Mode

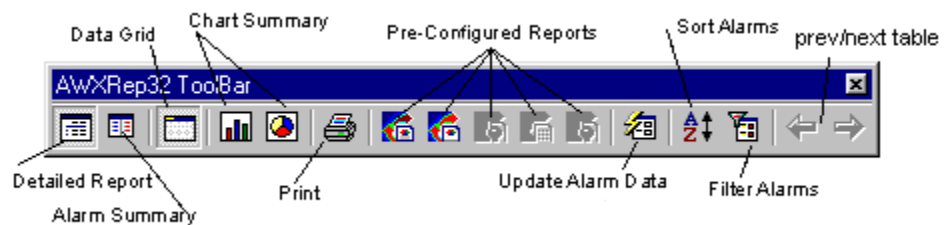
To enter Runtime, either click the 'Enter Runtime' icon from the toolbar or select **Actions - Runtime Mode** from the main menu. Once in Runtime the following data grid will appear.

AWXRep32 - [EventLog]			
	EventTime	Tag	Message
	12/3/99 1:09:13 PM	Scale	Reading on the sca
	12/3/99 1:09:13 PM	Tank1	The level in tank1 is
	12/3/99 1:09:13 PM	Ash Content	Ash Content of Tar
	12/3/99 1:09:13 PM	Box Line	Belt1 on the Box Li
	12/3/99 1:09:13 PM	Coolant Level	Warp core breach
	12/3/99 1:09:13 PM	Pump1	Pressure in pump1
	12/3/99 1:09:13 PM	Alkaline Level	Alkaline level in Tar
	12/3/99 1:09:25 PM	Coolant Level	Coolant leak detect
	12/3/99 1:09:41 PM	Humidity	Core humidity is hig
	12/3/99 1:09:45 PM	Tank PSI	The PSI in Tank1 is
	9/19/00 12:23:19 PM	AWXSvr32	ICONICS AlarmWor
	12/3/99 1:09:45 PM	Humidity	Core humidity is too
	12/3/99 1:09:45 PM	Belt Speed	Belt Speed of Pump

Runtime Mode

Note: If Runtime Mode does not show any data make sure you have opened the data by selecting 'Data Source' on the General Tab in the Properties Pages while in Configuration Mode. Also, if you deleted one of the three default columns (EventTime, Tag, or Message), during Logger Configuration Runtime will not run.

Using the AWXRep32 ToolBar the user is able to execute certain operations and change how data will be viewed. Double click anywhere in the ActiveX control's client box while in Runtime Mode to show the AWXRep32 ToolBar.



AWXRep32 Toolbar

There are two different views for the user to choose from: Data Grid or Chart Summary. When viewing Data Grid, the user can further choose from a

detailed report or a summary report. These views are described below along with the other tool button options.

Data Grid

Chart Summary

Printing Current Alarms

Choosing a Crystal Report Template

Updating Alarm Data

Adjust Columns width

Sorting and Filtering using AWXRep32 ToolBar

Data Grid

By default the Runtime screen will display in **Data Grid - Details** view. The Data Grid will give the user all data for the alarms in the report. The reported elements are determined by the columns selected during configuration. Additionally, the user may be able to change the view slightly by changing the width of the columns if this option was enabled during configuration.

If the user selects the 'Summary' button the view will change to display only the Tag name and the Alarm Summary. The data will be listed from highest to lowest number of alarms.

Chart Summary

If the user selects the 'Chart Summary' button a chart similar to the following figure will display.

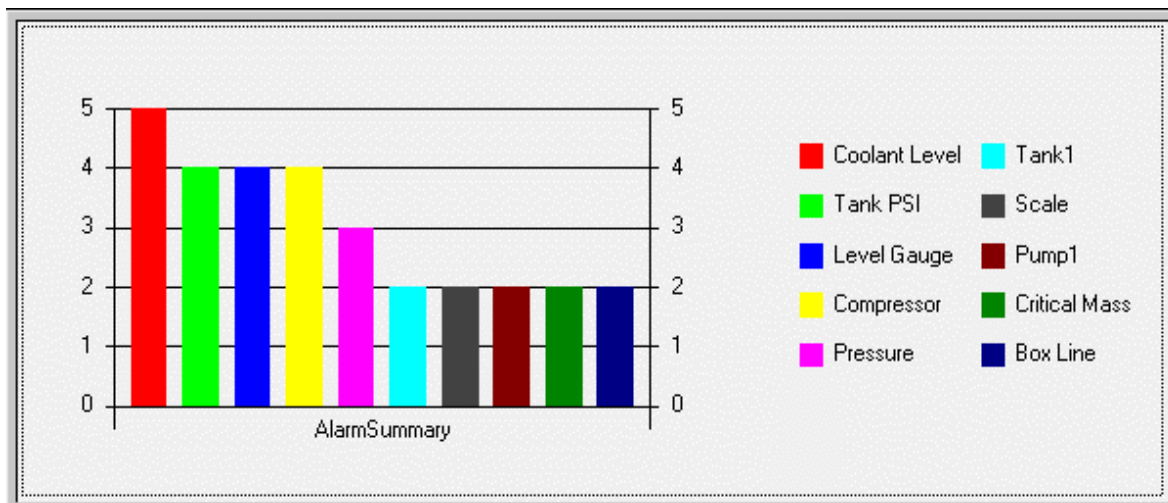
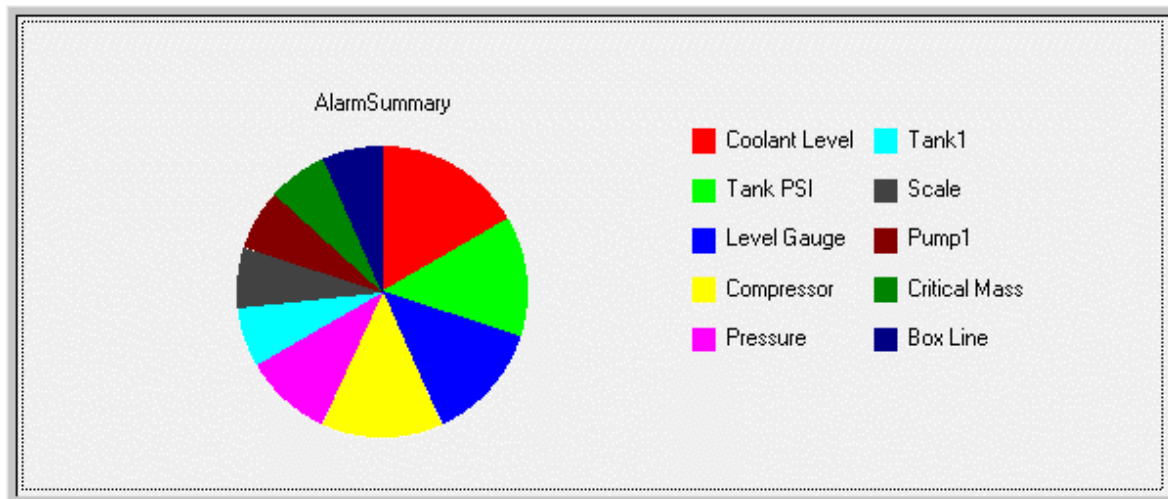


Chart Summary

This chart shows how many alarms of every type are present in the Log that is being reported. The number of items charted depends on the options selected on the Filtering tab. If there are no options selected to filter by all alarm types will be charted. The data will be shown from highest to lowest number of alarms for easiest viewing. The above figure shows the bar type of chart. This type is helpful if the user wishes to see how many alarms of each type there were. Alternatively, the user can select to view a Pie type chart, which will show what percentage of each alarm type exists in the report.



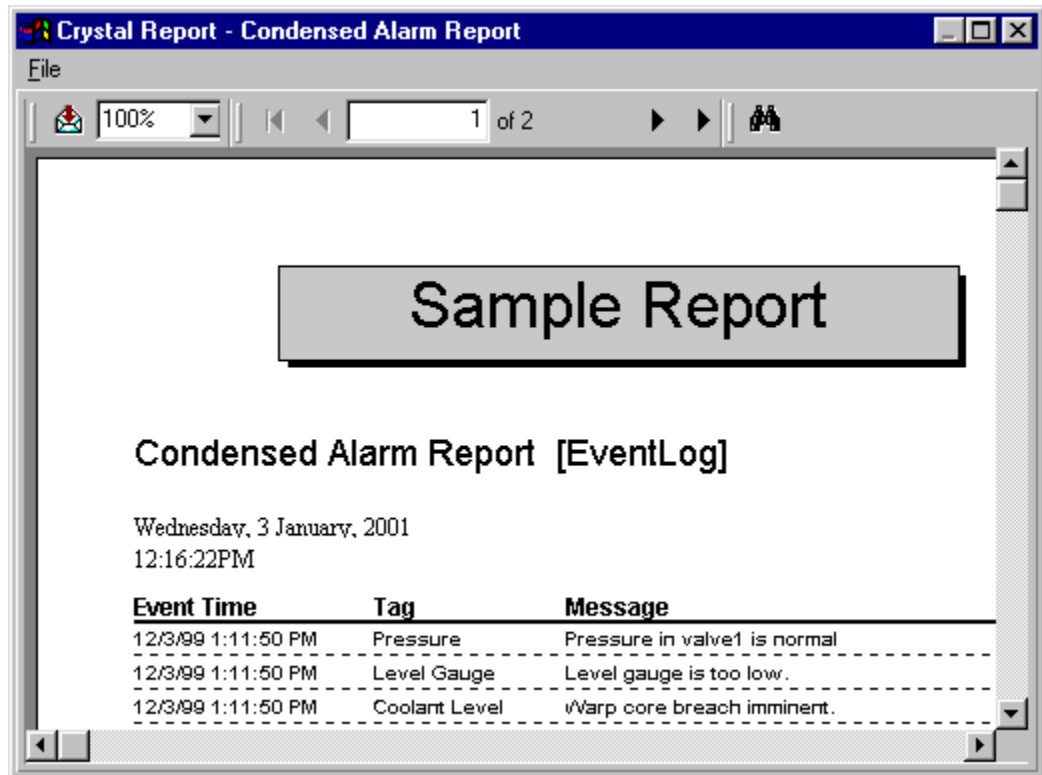
Pie Chart Summary

Printing Current Alarms

The 'Print Current Alarms' button on the AWXRep32 Toolbar will bring up the standard Print dialog box where you can choose your printing settings. The current view you have showing will print with the date, time, description, and page number at the top of the page.

Choosing a Crystal Report Template

If you have configured any reporting templates on the 'Reporting' tab in configuration mode, you will see one to five reports on the toolbar from which to choose. Clicking on a report button will launch a Crystal Reports viewer and show the selected report. From there, all of the features of Crystal Reports are at your disposal for printing, formatting and saving your report. A sample report is shown below.



Sample Report

Updating Alarm Data

There is a button on the AWXRep32 Toolbar that will update the data shown with the most current alarm data.

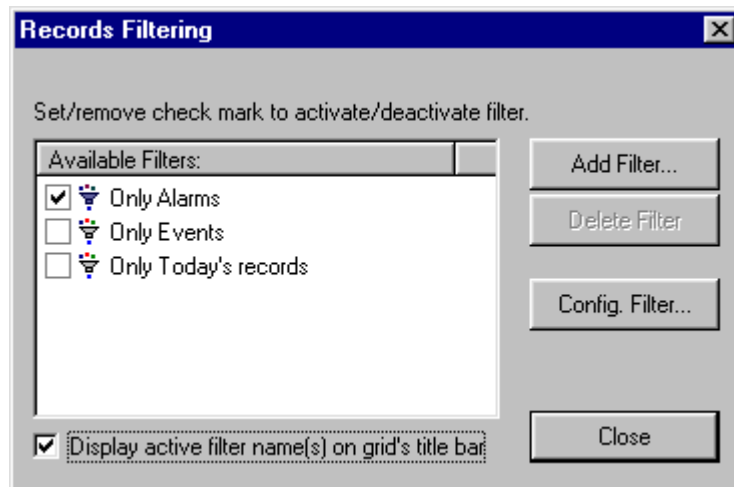
Adjust Columns Width

You can adjust each column independently by moving the cursor over the vertical line in-between the columns (a double arrow will appear) and moving the line while holding down the mouse button.

Sorting and Filtering using AWXRep32 ToolBar

The last two buttons before the left and right arrows on the AWXRep32 Toolbar are used to sort and filter the alarm data. The 'Sort Alarms' button will bring you to the dialog box described under **Records Sorting**.

The 'Filter Alarms' button will bring up the following dialog box. Adding, removing and configuring filters are described under **Record Filtering**.



Alarm Filtering

AlarmWorX32

Viewer OLE Automation Reference



Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.
Foxborough, MA 02035

Tel: 508-543-8600

FAX: 508-543-1503

E-MAIL: support@iconics.com

WEB: <http://www.iconics.com>

Contents

Introduction

Automation Interfaces

Available Control Properties

Available Control Methods

Available Control Events

Introduction

This reference describes the OLE Automation features available in the AlarmWorX32 Viewer.

Automation Interfaces

The AlarmWorX32 Viewer provides a COM interface that allows automation interfaces run from within the ActiveX container to manipulate the Viewer as it is running. The interface is available to all programming languages that support COM including Visual Basic (VB), Visual Basic for Applications (VBA) and Microsoft Visual C⁺⁺.

To access the Automation interface from VB and VBA, AlarmWorX32 Viewer must be made available by choosing **Project - Components** from the main menu in the VB or VBA development environment and selecting **AWXVIEW32 ActiveX** in the list of available components.

Available Control Properties

boolean Check

Description

Enable/Disable URL file support.

Remarks

If check is True URL file support is enabled. FileEnabled must also be True. URL file support is only available for loading.

Example

AWXVIEW320CX1.Check = True

boolean FileEnabled

Description

Enable/Disable file storage

Remarks

If FileEnabled is True file support is enabled. File support consists of loading and saving configuration information to a file other than the container document. It is recommended that the LoadSaveFile Method be used instead.

Example

AWXVIEW320CX1.FileEnabled = True

BSTR Caption

Description

String used for Title

Remarks

This property has no function at this time.

Example

BSTR Filename

Description

Name of the file to load/save

Remarks

If FileEnabled is True file support is enabled. The Filename is the name of the file and the path of the file to load and save configuration information to. File names must have an awv extension. It is recommended that the LoadSaveFile Method be used instead.

Example

Sets the file name

```
AWXVIEW320CX1.Filename = "C:\test.awv"
```

Gets the file name

```
Dim Name as String  
Name = AWXVIEW320CX1.Filename
```

BSTR UrlPath

Description

Name of the URL path and file to load from

Remarks

The UrlPath is the name of the file and the path of the file to load configuration information from. File names must have an awv extension. Please note that configuration information can not be saved to a URL file.

Example

Sets the URL file name

```
AWXVIEW320CX1.UrlPath = "http:\12.13.150.2\test.awv"
```

Gets the URL file name

```
Dim Name as String  
Name = AWXVIEW320CX1.UrlPath
```

boolean AutoSize

Description

Enable/Disable AutoSize as default sizing method.

Remarks

AutoSize determines the base size of a column based on the size of the header text of that column.

Example

```
AWXVIEW320CX1.AutoSize = True
```

boolean DispRow

Description

Enable/Disable displaying row header

Remarks

Displaying the row header enables an operator to change the row width size during runtime. This property can be used to with the NumRowCheck property to provide a numerical value for each row.

Example

```
AWXVIEW320CX1.DispRow = True
```

boolean ManSize

Description

Enable/Disable Manual Size as default size

Remarks

Enabling this property will prevent the viewer from altering the sizes of any columns except for the last column. Use this property if you wish to manually set base column sizes.

Example

```
AWXVIEW320CX1.AutoSize = False
```

boolean NumRowCheck

Description

Enable/Disable numbering row header. Only enabled when DispRow is Enabled.

Remarks

Use this property in conjunction with the DispRow property to display a number for each alarm line.

Example

Example displays the number of each alarm line

```
AWXVIEW320CX1.DispRow = True  
AWXVIEW320CX1.NumRowCheck = True
```

boolean Scale

Description

N/A

Remarks

This property has no function at this time.

Example

boolean Title

Description

N/A

Remarks

This property has no function at this time.

Example

boolean WrapText

Description

Enable/Disable text wrapping. Only visible when row size is greater than 1.

Remarks

Enabling this property causes the all column text to wrap to the next line. The wrap effect is contained within each individual column cell. Please note that this effect is only visible if the RowSize property is greater than 1 and the text information is larger than one column length.

Example

This example sets the Column height to 2 and enables text wrapping.

```
AWXVIEW320CX1.WrapText = True  
AWXVIEW320CX1.RowSize = 2
```

long MaxLength

Description

Sets default cell length.

Remarks

This property will set all columns in the viewer to the sizes specified (excluding the last column). Use this property if you wish to set all base column sizes to the same size.

Example

```
AWXVIEW320CX1.MaxLength = 10
```

JUSTIFY BaseJustify

Description

Sets default justification.

Remarks

This property sets the base justification for each column. It makes use of the enumerated values JustifyCenter, JustifyLeft, JustifyRight. The second half of the enumerated value name denotes they type of justification. For example JustifyCenter would center the text for each column.

Example

```
AWXVIEW320CX1.BaseJustify = JustifyCenter
```

long RowSize

Description

Sets base Row height.

Remarks

This property directly effect how much of the screen space is taken up for each row.

Example

This example sets the Column height to 2.

```
AWXVIEW320CX1.RowSize = 2
```

OLE_COLOR BackColor

Description

Sets the default background color

Remarks

This color is used for the base background color of each row (excluding row and column headers).

RGB defines colors in the order of red, green, blue. A good trick to use when picking colors is to open a graphics package and use the color palate to determine the RGB numbers. This will allow a visual representation of the color and provide the exact corresponding RGB values.

Example

```
AWXVIEW320CX1.BackColor = RGB(255, 255, 255)
```

OLE_COLOR HeaderColor

Description

Sets the default Header color

Remarks

RGB defines colors in the order of red, green, blue. A good trick to use when picking colors is to open a graphics package and use the color palate to determine the RGB numbers. This will allow a visual representation of the color and provide the exact corresponding RGB values.

Example

```
AWXVIEW320CX1.HeaderColor = RGB(255, 255, 255)
```

OLE_COLOR TextColor

Description

Sets the default text color

Remarks

RGB defines colors in the order of red, green, blue. A good trick to use when picking colors is to open a graphics package and use the color palate to determine the RGB numbers. This will allow a visual representation of the color and provide the exact corresponding RGB values.

Example

```
AWXVIEW320CX1.TextColor = RGB(255, 255, 255)
```

OLE_COLOR GridColor

Description

Sets the default background color

Remarks

This color is used for all grid space not covered by a row.

RGB defines colors in the order of red, green, blue. A good trick to use when picking colors is to open a graphics package and use the color palate to determine the RGB numbers. This will allow a visual representation of the color and provide the exact corresponding RGB values.

Example

```
AWXVIEW320CX1.BackColor = RGB(255, 255, 255)
```

GRIDLINE HStyle

Description

Sets horizontal grid style

Remarks

Grid styles other than solid can not have widths greater than 1. Also some grid styles, other than solid, have trouble printing on some printers.

GRIDLINE is an enumerated list containing the following values:
GridLineDash, GridLineDashdot, GridLineDashdotdot, GridLineDot,
GridLineSolid.

Example

Example of a valid grid style and width settings

```
AWXVIEW320CX1.HStyle = GridLineSolid  
AWXVIEW320CX1.HWidth = 2
```

```
AWXVIEW320CX1.HStyle = GridLineDot
```

AWXVIEW320CX1.HWidth = 1

GRIDLINE VStyle

Description

Sets vertical grid style

Remarks

Grid styles other than solid can not have widths greater than 1. Also some grid styles, other than solid, have trouble printing on some printers.

GRIDLINE is an enumerated list containing the following values:
GridLineDash, GridLineDashdot, GridLineDashdotdot, GridLineDot,
GridLineSolid.

Example

Example of a valid grid style and width settings

```
AWXVIEW320CX1.VStyle = GridLineSolid  
AWXVIEW320CX1.VWidth = 2
```

```
AWXVIEW320CX1.VStyle = GridLineDot  
AWXVIEW320CX1.VWidth = 1
```

boolean HCheck

Description

Enable/Disable horizontal grid lines

Example

```
AWXVIEW320CX1.HCheck = True
```

boolean VCheck

Description

Enable/Disable vertical grid lines

Example

```
AWXVIEW320CX1.VCheck = True
```

long HWidth

Description

Sets horizontal grid line width

Remarks

Grid styles other than solid can not have widths greater than 1. Also some grid styles, other than solid, have trouble printing on some printers.

Example

Example of a valid grid style and width settings

```
AWXVIEW320CX1.HStyle = GridLineSolid  
AWXVIEW320CX1.HWidth = 2
```

```
AWXVIEW320CX1.HStyle = GridLineDot  
AWXVIEW320CX1.HWidth = 1
```

long VWidth

Description

Sets vertical grid line width

Remarks

Grid styles other than solid can not have widths greater than 1. Also some grid styles, other than solid, have trouble printing on some printers.

Example

Example of a valid grid style and width settings

```
AWXVIEW320CX1.VStyle = GridLineSolid  
AWXVIEW320CX1.VWidth = 2
```

```
AWXVIEW320CX1.VStyle = GridLineDot  
AWXVIEW320CX1.VWidth = 1
```

OLE_COLOR HorColor

Description

Sets horizontal grid line color

Remarks

RGB defines colors in the order of red, green, blue. A good trick to use when picking colors is to open a graphics package and use the color palate to determine the RGB numbers. This will allow a visual representation of the color and provide the exact corresponding RGB values.

Example

```
AWXVIEW320CX1.HorColor = RGB(255, 255, 255)
```

OLE_COLOR VerColor

Description

Sets vertical grid line color

Remarks

RGB defines colors in the order of red, green, blue. A good trick to use when picking colors is to open a graphics package and use the color palate

to determine the RGB numbers. This will allow a visual representation of the color and provide the exact corresponding RGB values.

Example

```
AWXVIEW320CX1.VerColor = RGB(255, 255, 255)
```

boolean AutoStart

Description

Enable/Disable AutoStart feature

Remarks

AutoStart allows the viewer to enter a runtime state when the container has AmbientUserMode set to True.

Example

```
AWXVIEW320CX1.AutoStart = True
```

boolean NoToolTips

Description

Enable/Disable NoToolTips as runtime option

Remarks

Setting this property to False will gray out the NoToolTips option on the right click runtime pop-up menu.

Example

```
AWXVIEW320CX1.NoToolTips = False
```

boolean Help

Description

Enable/Disable Help ToolTips as runtime option

Remarks

Setting this property to False will gray out the Help option on the right click runtime pop-up menu.

Example

```
AWXVIEW320CX1.Help = False
```

boolean Event

Description

Enable/Disable Event ToolTips as runtime option

Remarks

Setting this property to False will gray out the Event option on the right click runtime pop-up menu.

Example

```
AWXVIEW320CX1.Event = False
```

boolean Context

Description

Enable/Disable Context ToolTips as runtime option

Remarks

Setting this property to False will gray out the Context option on the right click runtime pop-up menu.

Example

```
AWXVIEW320CX1.Context = False
```

boolean Comment

Description

Enable/Disable Comment ToolTips as runtime option

Remarks

Setting this property to False will gray out the Comment option on the right click runtime pop-up menu.

Example

```
AWXVIEW320CX1.Comment = False
```

boolean AlarmLine

Description

Enable/Disable entire alarm line popup ToolTips as runtime option

Remarks

Setting this property to False will gray out the AlarmLine option on the right click runtime pop-up menu.

Example

```
AWXVIEW320CX1.AlarmLine = False
```

TOOLTIP DefaultTooltip

Description

Set default ToolTip setting for runtime.

Remarks

This property sets ToolTip to use on startup. Valid ToolTip values are as follows:

- PopupAlarmLine
- PopupComment
- PopupContext
- PopupHelp
- PopupNone

Example

```
AWXVIEW32OCX1.Comment = False
```

short TooltipWidth

Description

Set tooltip width.

Remarks

Longer ToolTip widths will cause the ToolTip to have fewer rows.

Example

```
AWXVIEW32OCX1.TooltipWidth = 15
```

long CommentHeader

Description

Defines a header to associate with the Comment pop-up ToolTips

Remarks

This call should be used in conjunction with the position of the desired header. This causes the comment option to display the information in the associated column.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW32OCX1.ColumnNametoPosition("Column1")  
AWXVIEW32OCX1.CommentHeader = Val1
```

long HelpHeader

Description

Defines a header to associate with the Help pop-up ToolTips

Remarks

This call should be used in conjunction with the position of the desired header. This causes the help option to display the information in the associated column.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.ColumnNametoPosition("Column1")  
AWXVIEW320CX1.HelpHeader = Val1
```

boolean AlarmDouble

Description

Enable/Disable left mouse double click to acknowledge alarms

Remarks

Example

```
AWXVIEW320CX1.AlarmDouble = False
```

boolean AckDialog

Description

Enable/Disable the forcing of alarm acknowledgement through ack dialog.

Remarks

Enabling this property forces all acknowledgements through the ack dialog. This differs from the AlarmDouble property in the fact that a double click with AckDailog set to true will bring up the acknowledge dialog.

Example

```
AWXVIEW320CX1.AckDialog = False
```

boolean EnableMoveCol

Description

Enable/Disable the changing of column order during runtime

Example

```
AWXVIEW320CX1.EnableMoveCol = True
```

boolean EnableResizeCol

Description

Enable/Disable the resizing of columns during runtime

Remarks

This feature should only be used for design and testing due to the fact that it would allow the user to completely hide a column of information.

Example

```
AWXVIEW320CX1.EnableResizeCol = True
```

boolean EnableResizeRow

Description

Enable/Disable the resizing of rows during runtime

Remarks

This feature should only be used for design and testing due to the fact that it would allow the user to completely hide alarm lines.

Example

```
AWXVIEW320CX1.EnableResizeRow = True
```

BSTR SelectedSource

Description

Returns the selected source string. Source name is chosen based on messages selected, alarm state, severity, and time

Remarks

This property will return the source string (tag name) that best fits the following criteria:

- 1) Highlighted alarm line.
- 2) Highest alarm state.
- 3) Highest severity.
- 4) Most recent time stamp.

Alarm state priority is defined as follows:

- 1) In Alarm and unacknowledged
- 2) In Alarm but acknowledged
- 3) Passed out of Alarm but unacknowledged

4) Tracking message

5) Operator message

Example

```
Dim SourceName As String  
SourceName = AWXVIEW320CX1.SelectedSource  
MsgBox (SourceName)
```

BSTR PrioritySource

Description

Returns the priority source string. Source name is chosen based on messages in alarm, severity, and time.

Remarks

This property returns a source name (tag name) that best fits the following criteria:

- 1) In alarm.
- 2) Highest severity.
- 3) Most recent time stamp.

If no alarm messages are in alarm and unacknowledged this property will return an empty string.

Example

```
Dim SourceName As String  
SourceName = AWXVIEW320CX1.PrioritySource  
MsgBox (SourceName)
```

Available Control Methods

boolean ShowColumn(COLIMNID ColumnId, boolean bShow)

Description

This method will hide and show columns. If bShow is True it will attempt to show the column. Returns a True on success.

Example

This example shows the column alarm type

```
Dim bVal as Boolean  
bVal = AWXVIEW320CX1.ShowColumn(ALARMTYPE, True)
```

boolean SetToolTip(TOOLTIP Id)

Description

This method will set the current Tooltip mode to the type associated with TOOLTIP id.

Remarks

See the DefaultTooltip property for TOOLTIP id list.

Example

```
Dim bVal as Boolean  
bVal = AWXVIEW320CX1.SetToolTip(PopupAlarmline)
```

boolean SortOn(short Count, COLIMNID Sort1, boolean Direction1, COLIMNID Sort2, boolean Direction2, COLIMNID Sort3, boolean Direction3, COLIMNID Sort4, boolean Direction4, COLIMNID Sort5, boolean Directon5)

Description

Sorts grid. Pass in columnID and direction combinations to set up sort. Count is the number of items to sort on. Pass zero for all unused parameters. The first pair can not be zero. Returns true on success. A direction of True causes accending order.

Remarks

The count parameter is used to determin how many columnID and direction pairs to use for sorting.

Example

This example causes the grid to be sorted first by alarm type and then by time. The effect is to have the earliest most recent alarms on top and the oldest operator messages on the bottom.

```
Dim bVal as Boolean
bVal = AWXVIEW320CX1.SortOn(2, ALARMTYPE, True, AEACTIONTIME, False , 0,
0, 0, 0, 0, 0)
```

boolean SetColumnName(BSTR Name, short Pos)

Description

This method sets the name of the column located at the position passed in.

Remarks

Column position can change due to runtime user interaction or script. Please be aware that column position counts hidden columns when doing calculations.

Example

```
Dim bVal As Boolean
bVal = AWXVIEW320CX1.SetColumnName("NewCol1", 3)
```

short ColumnNameToPosition(BSTR Name)

Description

Pass in string name (Case sensitive) returns column position. A return of zero is a failure.

Remarks

Column position can change due to runtime user interaction or script. Please be aware that column position counts hidden columns when doing calculations.

Example

```
Dim ColPos as Integer
ColPos = AWXVIEW320CX1.ColumnNameToPosition("Column1")
```

short ColumnNameToId(BSTR Name)

Description

Pass in string name (Case sensitive) returns column id. A Value of zero is a failure.

Remarks

The column id value corresponds to the following list:

AECHANGEMASK = 0,

AENEWSTATE = 1,

AESOURCE = 2,

AETIME = 3,

AEMESSAGE = 4,
AEEVENTTYPE = 5,
AEEVENTCATEGORY = 6,
AESEVERITY = 7,
AECONDITIONNAME = 8,
AESUBCONDITIONNAME = 9,
AEQUALITY = 10,
AEACKREQUIRED = 11,
AEACTIVETIME = 12,
AECOOKIE = 13,
AENUMEVENTATTRS = 14,
AEACTORID = 15,
ATTRIB1 = 16,
ATTRIB2 = 17,
ATTRIB3 = 18,
ATTRIB4 = 19,
ATTRIB5 = 20,
ATTRIB6 = 21,
ATTRIB7 = 22,
ATTRIB8 = 23,
ATTRIB9 = 24,
ATTRIB10 = 25,
ATTRIB11 = 26,
ATTRIB12 = 27,
ATTRIB13 = 28,
ATTRIB14 = 29,

ATTRIB15 = 30,
ATTRIB16 = 31,
ATTRIB17 = 32,
ATTRIB18 = 33,
ATTRIB19 = 34,
ATTRIB20 = 35,
ALARMTYPE = 36,

Example

```
Dim ColId as Integer  
ColId = AWXVIEW320CX1.ColumnNameToId("Column1")
```

BSTR IdtoColumnName(COLIMNID Id)

Description

Pass in columnID. Returns string name.

Remarks

See ColumnNameToId function for Id list.

Example

```
Dim ColName as String  
ColName = AWXVIEW320CX1.IdtoColumnName(AEACTIVETIME)
```

BSTR PositiontoColumnName(short Position)

Description

Pass in position. Returns string name.

Remarks

Column position can change due to runtime user interaction or script. Please be aware that column position counts hidden columns when doing calculations.

Example

```
Dim ColName as String  
ColName = AWXVIEW320CX1.PositiontoColumnName(2)
```

long GetNumberOfAlarms()

Description

Returns current number of items in alarm.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfAlarms()
```

long GetNumberOfAked()

Description

Returns current number of items in acknowledged state.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfAked()
```

long GetNumberOfUnacked()

Description

Returns current number of items in unacknowledged state.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfUnacked()
```

long GetNumberOfTracking()

Description

Returns current number of tracking items.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfTracking()
```

long GetNumberOfOperator()

Description

Returns current number of operator items.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfOperator()
```

long GetTotalNumberOfMessages()

Description

Returns total number of current messages.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetTotalNumberOfMessages()
```

long GetNumberOfFilterAlarms()

Description

Returns current number of items in alarm that pass through the client filter.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfFilterAlarms()
```

long GetNumberOfFilterAcked()

Description

Returns current number of items in acknowledged state that pass through the client filter.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfFilterAcked()
```

long GetNumberOfFilterUnacked()

Description

Returns current number of items in unacknowledged state that pass through the client filter.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfFilterUnacked()
```

long GetNumberOfFilterTracking()

Description

Returns current number of tracking items that pass through the client filter.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfFilterTracking()
```

long GetNumberOfFilterOperator()

Description

Returns current number of operator items that pass through the client filter.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfFilterOperator()
```

long GetTotalNumberOfFilterMessages()

Description

Returns total number of current messages that pass through the client filter.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.GetNumberOfFilterMessages()
```

short PositiontoId(short Pos)

Description

Pass in position returns column ID.

Remarks

Column position can change due to runtime user interaction or script. Please be aware that column position counts hidden columns when doing calculations.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.PositiontoId(2)
```

short IdtoPosition(COLIMNID Id);

Description

Pass in column ID return position.

Remarks

Column position can change due to runtime user interaction or script. Please be aware that column position counts hidden columns when doing calculations.

Example

```
Dim Val1 As Integer  
Val1 = AWXVIEW320CX1.IdtoPosition(AEACTIVETIME)
```

boolean SizeColAtPosition(short Pos, long Size);

Description

Pass in position and new size, changes size of column at that position. Return of true means success. Size must be larger than minimum width as seen in property pages. (equal to or larger than 1)

Remarks

Column position can change due to runtime user interaction or script. Please be aware that column position counts hidden columns when doing calculations. Width values are based on an internal algorithm composed of font size and type

Example

Changes column in position 2 to a width of 15

```
Dim bVal As Boolean  
bVal = AWXVIEW320CX1.SizeColAtPosition(2,15)
```

boolean SizeColWithId(COLIMNID Id, long Size)

Description

Pass in column ID and new size, changes size of column at that position. Return of true means success. Size must be larger than minimum width as seen in property pages. (equal to or larger than 1)

Remarks

Width values are based on an internal algorithm composed of font size and type

Example

Changes the time column to a width of 15

```
Dim bVal As Boolean  
bVal = AWXVIEW320CX1.SizeColWithId(AEACTIVETIME,15)
```

boolean MovCol(short FromStartPos, short FromEndPos, short ToPos)

Description

Move columns. FromStartPos is first column selected. FromEndPos is last column selected. Moves group of columns to ToPos. Cannot move past last column. Return of True means success.

Remarks

Please be aware that column position counts hidden columns when doing calculations.

Example

This Example will move columns 3, 4, and 5 in front of column 2.

```
Dim bVal As Boolean  
bVal = AWXVIEW320CX1.MovCol(3,5,2)
```

boolean EnableColTextColorOverride(COLIMNID Id, boolean Enable)

Description

Changes text color for specified column. Enable = true enables override. Returns true on success.

Example

This Example enables the text color of the time column to be overwritten and then changes the text color.

```
Dim bVal As Boolean
bVal = AWXVIEW320CX1.EnableColTextColorOverride(AEACTIVETIME, True)
Call AWXVIEW320CX1.SetColTextColor(AEACTIVETIME, RGB(255, 255, 255))
```

boolean EnableColBackColorOverride(COLIMNID Id, boolean Enable)

Description

Changes background color for specified column. Enable = true enables override. Returns true on success.

Example

This Example enables the background color of the time column to be overwritten and then changes the background color.

```
Dim bVal As Boolean
bVal = AWXVIEW320CX1.EnableColBackColorOverride(AEACTIVETIME, True)
Call AWXVIEW320CX1.SetColBackColor(AEACTIVETIME, RGB(255, 255, 255))
```

void SetColTextColor(COLIMNID Id, OLE_COLOR Color)

Description

Sets text color of specified column. Automatically enables text color override.

Remarks

This function will enable the text color to be overwritten.

Example

```
Dim bVal As Boolean
Call AWXVIEW320CX1.SetColTextColor(AEACTIVETIME, RGB(255, 255, 255))
```

void SetColBackColor(COLIMNID Id, OLE_COLOR Color)

Description

Sets background color of specified column. Automatically enables background color override.

Remarks

This function will enable the background color to be overwritten.

Example

```
Dim bVal As Boolean
Call = AWXVIEW320CX1.SetColBackColor(AEACTIVETIME, RGB(255, 255, 255))
```

OLE_COLOR GetColBackColor(COLIMNID Id)

Description

Returns text color of specified column. - **Note:** this is the override color.

Remarks

This returns the column background color not the base default background color.

Example

```
Dim rcolor As OLE_COLOR
rcolor = AWXVIEW320CX1.GetColBackColor(AEACTIVETIME)
```

OLE_COLOR GetColTextColor(COLIMNID Id)

Description

Returns background color of specified column. - **Note:** this is the override color.

Remarks

This returns the column text color not the base default text color.

Example

```
Dim rcolor As OLE_COLOR
rcolor = AWXVIEW320CX1.GetColTextColor(AEACTIVETIME)
```

boolean EnableRowTextColor(MESSAGETYPE wType, long LowSeverity, boolean Enable)

Description

Enables/Disables text color override for associated message type and rowID. Enable = True enables override. Returns true on success.

Remarks

The messagetype values are as follows:

- Alarm = 1
- Acknowledge = 2
- Unacknowledge = 3
- Operator = 4
- Tracking = 5
- Normal = 6

Example

This Example will disable the setting of the row text color for all alarm lines in alarm for severity values starting at 400 to the next configured severity or 1000. Which ever is less.

Dim bVal As Boolean

```
bVal = AWPVIEW320CX1.EnableRowTextColor(Alarm, 400, False)
```

boolean EnableRowBackColor(MESSAGETYPE wType, long LowSeverity, boolean Enable)

Description

Enables/Disables background color override for associated message type and rowID. Enable = True enables override. Returns true on success.

Remarks

See EnableRowTextColor method for available messagetype values.

Example

This Example will disable the setting of the row background color for all alarm lines in alarm for severity values starting at 400 to the next configured severity or 1000. Which ever is less.

Dim bVal As Boolean

```
bVal = AWPVIEW320CX1.EnableRowBackColor(Alarm, 400, False)
```

void SetRowTextColor(MESSAGETYPE wType, long LowSeverity, OLE_COLOR Color)

Description

Sets text color associated with message type and rowID. Automatically enables text color override.

Remarks

See EnableRowTextColor method for available messagetype values.

Example

This Example will change the row text color for all alarm lines in alarm for severity values starting at 400 to the next configured severity or 1000. Which ever is less.

```
Call AWPVIEW320CX1.SetRowTextColor(Alarm, 400, RGB(255, 255, 255))
```

void SetRowBackColor(MESSAGETYPE wType, long LowSeverity, OLE_COLOR Color)

Description

Sets background color associated with message type and rowID. Automatically enables background color override.

Remarks

See EnableRowTextColor method for available messagetype values.

Example

This Example will change the row background color for all alarm lines in alarm for severity values starting at 400 to the next configured severity or 1000. Which ever is less.

```
Call AWXVIEW320CX1.SetRowBackColor(Alarm, 400, RGB(255, 255, 255))
```

OLE_COLOR GetRowTextColor(MESSAGETYPE wType, long LowSeverity)

Description

Returns the row text color of the associated message type rowID combination. **Note** - this is the override color.

Remarks

See EnableRowTextColor method for available messagetype values.

Example

This example will get the row text color value for severity values starting at 400 to the next configured severity or 1000. Which ever is less.

```
Dim rcolor As OLE_COLOR  
rcolor = AWXVIEW320CX1.GetRowTextColor(Alarm, 400)
```

OLE_COLOR GetRowBackColor(MESSAGETYPE wType, long LowSeverity)

Description

Returns the row background color of the associated message type rowID combination. **Note** - this is the override color.

Remarks

See EnableRowTextColor method for available messagetype values.

Example

This example will get the row background color value for severity values starting at 400 to the next configured severity or 1000. Which ever is less.

```
Dim rcolor As OLE_COLOR  
rcolor = AWXVIEW320CX1.GetRowBackColor(Alarm, 400)
```

GlobalAck(BSTR comment)

Description

Acknowledges all received alarms.

Remarks

The comment is passed to the server.

Example

```
Call AWXVIEW320CX1.GlobalAck("down for maintenance")
```

AckSelectedRows(BSTR comment)

Description

Acknowledges rows selected by the operator.

Remarks

The comment is passed to the server.

Example

```
Call AWXVIEW320CX1.AckSelectedRows("down for maintenance")
```

void AckDisplayed(BSTR comment)

Description

Acknowledges all displayed alarms

Remarks

This function will acknowledge all alarm messages visible on the screen.
The comment is passed to the server.

Example

```
Call AWXVIEW320CX1.AckDisplayed("down for maintenance")
```

void AckFiltered(BSTR comment)

Description

Acknowledges all alarms that pass through the client filter. This differs from the global acknowledge by the fact that the global acknowledge will also acknowledge requested alarms that the client filter has filtered out.

Remarks

The comment is passed to the server.

Example

```
Call AWXVIEW320CX1.AckFiltered("down for maintenance")
```

void SetActorID(BSTR Operator, BSTR Station)

Description

Creates string of defining user who acknowledged an alarm. Syntax is as follows user: Operator Node: Station

Remarks

Once ActorID is called the ocx will not get the ActorID from the security system unless resetActorId is called.

Example

Call AWXVIEW32OCX1.SetActorID("NewOperator","NewStation")

void SetSecurity(long Security)

Description

Allows the AlarmWorX viewer security to be set through an automation interface.

Remarks

Once enabled, Iconics security server will not be used for the life of the ocx. The security map is as follows:

m_bSecAckPoint	0x0001
m_bSecGlobalAck	0x0002
m_bSecPageAck	0x0004
m_bSecScreenAck	0x0008
m_bSecAreaAck	0x0010
m_bSecVBEvent	0x0020
m_bSecSort	0x0040
m_bSecFilter	0x0080
m_bSecDetails	0x0100
m_bSecProperty	0x0200
m_bSecApplication	0x0400
m_bSecDisplay	0x0800
m_bSecTooltip	0x1000
m_bSecColumn	0x2000
m_bSecLanguage	0x4000
m_bSecChFilter	0x8000

A 1 enables the feature and a 0 disables the feature. By default all items are disabled.

A registry setting of Security 0 must be added to
HKEY_LOCAL_MACHINE\software\iconics\lawxview32 for this automation

method to be enabled. This registry setting disables the use of the security server for the Alarm viewer.

Example

This Example enables all options except AckPoint (point acknowledge)

```
Dim Security As Integer
Security = 65534 '65534 = 0xffff
Call AWXVIEW320CX1.SetSecurity(Security)
```

void ResetActorID()

Description

Sets the Actor ID back to the internal actor id string.

Example

```
Call AWXVIEW320CX1.ResetActorID()
```

void FilterOff()

Description

Turns off all client filters.

Example

```
Call AWXVIEW320CX1.ResetActorID()
```

BSTR GetActiveFilter()

Description

Returns the active client filter name.

Remarks

This is the name of the filter not the filter expression

Example

```
Dim FilterName As String
FilterName = AWXVIEW320CX1.GetActiveFilter()
```

BSTR GetFilter(LPCTSTR Filter)

Description

Returns the filter expression.

Remarks

Given the name of an existing filter this method will return the filter expression string. An empty string is returned if the filter does not exist.

Example

```
Dim FilterName As String
Dim FilterExpression As String
FilterExpression = AWXVIEW320CX1.GetFilter(FilterName)
```

boolean SetFilter(LPCTSTR Filter, BSTR Expression)

Description

Creates Filter if Filter does not already exist. Sets Expression as the expression string associated with Filter. Returns a true on success.

Remarks

The best way to ensure accurate expression syntax is to create the string using the client filter dialog and then copy the string expression. This method only adds the filter and filter name to the list of possible client filters. It does not activate the filter.

Example

This example creates a filter named filter1 with an expression that only passes alarms with a source name beginning with Tag.

```
Dim FilterName As String
Dim FilterExpression As String

FilterName = "Filter1"
'actual expression x= like({{Source}}, $"Tag*"$, 0)
FilterExpression = "like({{Source}}, $"Tag*"$, 0)"
FilterExpression = AWXVIEW320CX1.SetFilter(FilterName, FilterExpression)
```

boolean EnableFilter(LPCTSTR Filter, boolean State)

Description

Turns Filter on if state is True or off if state is False. Returns true on success.

Remarks

This method will turn filters on and off. When a filter is turned on, the previous filter will be turned off. It is only possible to have one filter running at any given time.

Example

This example turns the filter named filter1 on.

```
Dim FilterName As String
Dim bVal as Boolean
FilterName = "Filter1"
bVal = AWXVIEW320CX1.EnableFilter(FilterName, True)
```

boolean AlarmFilter(BSTR Filter, long StartSeverity, long EndSeverity, long AlarmType, BSTR AttributeField1, BSTR Attribute1, BSTR AttributeField2, BSTR Attribute2)

Description

Sets Filter with the following syntax:

```
( {{Severity}} >= StartSeverity )&&  
( {{Severity}} <= EndSeverity )&&  
( {{AlarmType}} == AlarmType )&&  
( like({{Attribute1}}, $"AttributeField1"$,0) )&&  
( like({{Attribute2}}, $"AttributeField2"$,0) )
```

Remarks

AlarmType decodes as follows:

- ALARM 1
- ACK 2
- UNACK 3
- OPER 4
- TRACK 5
- NORM 6

An AlarmType of 7 or greater removes alarm type from the filter.

The attribute field compare the string AttributeField with the string found in the column of the associated attribute. Comparison is done using the like function. The like function behaves identically to the VB like operator.

Attribute1 is removed from the filter if AttributeField1 is the string *

Attribute2 is removed from the filter if AttributeField2 is the string *

It is recommended that the SetFilter and EnableFilter methods are used instead of this method.

Example

```
Dim bVal as Boolean  
bVal = AWXVIEW320CX1.AlarmFilter("Filter1",0,1000, Alarm,  
"","extracol1","","extracol2")
```

void EnterAnimate()

Description

Causes the ActiveX to enter runtime mode.

Example

This example causes the viewer to exit runtime, load file test.awv, and enter runtime.

```
Dim FileName As String
Dim bVal as Boolean

FileName = "c:\test\test.awv"
Call AWXVIEW320CX1.ExitAnimate()
BVal = AWXVIEW320CX1.LoadSaveFile(FileName,1)
Call AWXVIEW320CX1.EnterAnimate()
```

void ExitAnimate()

Description

Causes the ActiveX to exit runtime mode and enter design mode.

Example

This example causes the viewer to exit runtime, load file test.awv, and enter runtime.

```
Dim FileName As String
Dim bVal as Boolean

FileName = "c:\test\test.awv"
Call AWXVIEW320CX1.ExitAnimate()
BVal = AWXVIEW320CX1.LoadSaveFile(FileName,1)
Call AWXVIEW320CX1.EnterAnimate()
```

boolean LoadSaveFile(BSTR FileName, short Load)

Description

Will cause the ActiveX to load or save an *.awv file. If Load is 1 then the file will be loaded. Load 0 will cause the file to be saved.

Remarks

This method will only load or save a file when the ActiveX is in design mode. The ActiveX properties Check must be False and FileEnabled must be True. This should be done through the property pages and saved as part of configuration. To set the property pages to load and save from a file, select Configuration File in the general properties tab and disable URL Path option. FileName should be the full path and filename including extension.

Example

This example causes the viewer to exit runtime, load file test.awv, and enter runtime.

```
Dim FileName As String
Dim bVal as Boolean

FileName = "c:\test\test.awv"
Call AWXVIEW320CX1.ExitAnimate()
BVal = AWXVIEW320CX1.LoadSaveFile(FileName,1)
Call AWXVIEW320CX1.EnterAnimate()
```

long ControlSize(long cx, long cy)

Description

Sets the size of the ocx. Returns a 0 on success.

Remarks

Cx and cy are in windows pixel units.

Example

```
Dim size As Integer  
size = AWXVIEW32OCX1.ControlSize(200, 300)
```

boolean ChangeNode(BSTR OldNode, BSTR NewNode)

Description

This method changes the node name associated with the alarm subscription. This method will cause the alarm subscription to switch the network nodes it is using for data collection.

Remarks

There are possible performance hits when using this function.

Example

```
Dim bVal As Boolean  
bVal = AWXVIEW32OCX1.ChangeNode("PC12", "PC13")
```

void UpdateDisplay()

Description

This method causes a redraw of the internal grid information.

Remarks

This should be used at the end of a script to force the update of automation color changes.

Example

```
Dim bVal As Boolean  
Call AWXVIEW32OCX1.SetColBackColor(AEACTIVETIME, RGB(255, 255, 255))  
Call AWXVIEW32OCX1.SetColTextColor(AEACTIVETIME, RGB(255, 255, 255))  
Call AWXVIEW32OCX1.UpdateDisplay()
```

BSTR SelectedColRowInfo(COLIMNID Id)

Description

This method returns a string containing the text in the column corresponding to 'Id' for the row selected in the Alarm Viewer ActiveX.

Example

```
Dim TimeInfo as string
'Get the time of the selected row
TimeInfo = AWXVIEW320CX.1.SelectColRowInfo(AETIME)
```

BSTR ColRowInfo(long Row, COLIMNID Id)

Description

This method takes the row number and column ID as input parameters. It returns the row and column information in the form of a string.

Example

```
Dim s As String
s = ThisDisplay.AWXVIEW320CX1.ColRowInfo(1, AESOURCE)
MsgBox (s)
```

This example will pop up a message box with the source of the alarm message.

Available Control Events

OnRightMouseClicked()

Description

Event posted on user right mouse click

Example

```
Private Sub AWXVIEW320CX1_OnRightMouseClicked()  
    'Your code here  
End Sub
```

OnLeftMouseClicked()

Description

Event posted on user left mouse click

Example

```
Private Sub AWXVIEW320CX1_OnLeftMouseClicked()  
    'Your code here  
End Sub
```

OnRightMouseDBLClick()

Description

Event posted on user right double mouse click

Example

```
Private Sub AWXVIEW320CX1_OnRightMouseDblClick()  
    'Your code here  
End Sub
```

OnLeftMouseDBLClick()

Description

Event posted on user left double mouse click

Example

```
Private Sub AWXVIEW320CX1_OnLeftMouseDblClick()  
    'Your code here  
End Sub
```

OnNewAlarm()

Description

Event posted when any new message enters the viewer.

Remarks

Please Note: This event occurs when any new OPC alarm and events message is passed into the viewer. This includes client filtered messages, operator messages, tracking messages, alarm messages, and acknowledge messages. No information provided to the event that denotes message type.

Example

```
Private Sub AWXVIEW320CX1_OnNewAlarm()  
    'Your code here  
End Sub
```

OnPopupEvent()

Description

Event posted when the user selects the VB event menu item from the right click pop-up menu.

Example

```
Private Sub AWXVIEW320CX1_OnPopupEvent()  
    'Your code here  
End Sub
```

AlarmWorX32

Logger OLE Automation Reference



Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.
Foxborough, MA 02035

Tel: 508-543-8600

FAX: 508-543-1503

E-MAIL: support@iconics.com

WEB: <http://www.iconics.com>

Contents

Introduction

Automation Interfaces

Available Control Properties

Introduction

This reference describes the OLE Automation features available in the AlarmWorX32 Logger.

Automation Interfaces

The AlarmWorX32 logger provides a COM interface that allows external programs to manipulate the logger as it is running. The interface is available to all programming languages that support COM including Visual Basic (VB), Visual Basic for Applications (VBA) and Microsoft Visual C⁺⁺.

To access the Automation interface from VB and VBA, AlarmWorX32 logger must be made available by choosing **Project - References** from the main menu in the VB or VBA development environment and selecting **AWXLog32 1.0 Type Library** in the list of available references.

Available Control Properties

LoggingEnabled

Description

True if logging to database is enabled. Applies to all running configurations that match the current **Filter Property**. Property is read/write.

Remarks

Values written to this property are transient. The value written is not stored in the configuration database and hence is "lost" when the logger is shut down.

Reading this property will throw an error in the following cases:

The logger has just started and the configuration has not been read yet.

Multiple running configurations match the filter and they have differing values (some True, some False).

Example

This example shows how to determine if logging is enabled.

```
Dim bEnabled as Boolean  
bEnabled = AWXLogger.LoggingEnabled
```

This example shows how to turn on/off logging

```
AWXLogger.LoggingEnabled = True
```

PrintingEnabled

Description

True if logging to printer is enabled. Applies to all running configurations that match the current **Filter Property**. Property is read/write.

Remarks

Values written to this property are transient. The value written is not stored in the configuration database and hence is "lost" when the logger is shut down.

Reading this property will throw an error in the following cases:

The logger has just started and the configuration has not been read yet.

Multiple running configurations match the filter and they have differing values (some True, some False).

Example

This example shows how to determine if printing is enabled.

```
Dim bEnabled as Boolean
bEnabled = AWXLogger.PrintingEnabled
```

This example shows how to turn on/off printing.

```
AWXLogger.PrintingEnabled = True
```

Count

Description

Returns the number of running configurations who's names match the current Filter Property. Property is read-only.

Remarks

Reading this property will throw an error if the logger has just started and the configuration has not been read yet.

Example

```
Dim n as Long
n = AWXLogger.Count
```

Filter

Description

Wildcard string that a running configuration name must match in order to be affected by other methods and properties of this object. String property is read/write.

Remarks

If the name of a running configuration does **not** match the filter, then that configuration is ignored by all other methods and properties of this object. The default initial value for this Property is "*" meaning all running configurations will match, hence by default, all off the methods and properties will operate simultaneously on all running configurations.

Example

```
Dim sFilter as String
sFilter = AWXLogger.Filter

'set filter for all configurations that begin with b
AWXLogger.Filter = "b*"
```

Index

Description

The name of a running configuration (that matches the **Filter Property**) for a given index value in the array (VB Collection).

Syntax

Index(long n)

Where **n** is an integer from 1 to **Count**

Remarks

The AlarmWorX32 logger Automation object is designed as a VB Collection. Each element in the collection is the name of an active configuration that matches the current **Filter Property**.

Example

These three examples do the same thing -- print the name of each configuration.

```
Dim AWXLogger As AWXLog32Auto
Set AWXLogger = New AWXLog32Auto

' method 1
For Each ConfigName In AWXLogger
    Print ConfigName
Next ConfigName

' method 2
For i = 1 To AWXLogger.Count
    Print AWXLogger(i)
Next i

' method 3
For i = 1 To AWXLogger.Count
    Print AWXLogger.Item(i)
Next i
```

iCONICS AlarmWorX32

Report OLE Automation Reference



Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.

Foxborough, MA 02035

Tel: 508-543-8600

FAX: 508-543-1503

E-MAIL: support@iconics.com

WEB: <http://www.iconics.com>

Contents

Introduction

Automation Interfaces

Available Control Properties

Available Control Methods

Introduction

This reference describes the OLE Automation features available in the AlarmWorX32 Report.

Automation Interfaces

The AlarmWorX32 Report provides a COM interface that allows automation interfaces run from within the ActiveX container to manipulate the Report ActiveX control as it is running. The interface is available to all programming languages that support COM, including Visual Basic (VB), Visual Basic for Applications (VBA), and Microsoft Visual C++.

To access the Automation interface from VB and VBA, AlarmWorX32 Report must be made available by choosing **Project > Components** from the main menu in the VB or VBA development environment and selecting **ICONICS AWXRep32 ActiveX** in the list of available components.

The description of every property or method in this reference is followed by the code example for typical use of that property or method. The example code is written in Visual Basic for Applications (VBA) language, built in the AlarmWorX32 container application (AwX32.exe). It assumes that AWXRep32 ActiveX is embedded within **AlarmWorX** display and its name property is set to "AWXRep321."

Available Control Properties

AWXRep32Title

Type: String

Description

Sets/gets title string on General property page.

Example

```
Dim strNewTitle as String  
ThisDocument.AWXRep321.AWXRep32Title=strNewTitle
```

AppearanceMode

Type: Long

Description

Determines the appearance of data grid. This property can be one of the following:

- 0 - flat look
- 1 - 3D look

Example

Sets 3D look for the grid.
`ThisDocument.AWXRep321.AppearanceMode=1`

BackColor

Type: OLE_COLOR

Description

Sets/gets grid background color.

Example

Sets grid background color to cyan.
`ThisDocument.AWXRep321.BackColor=RGB(0, 255, 255)`

BorderType

Type: String

Description

Sets/gets control's border appearance. Can be one of the following:

"0" - none

"1" - flat border

"2" - 3D border

Example

Sets 3D look for the control's border.

```
ThisDocument.AWXRep321.BorderType="2"
```

ChartBackColor

Type: OLE_COLOR

Description

Sets/gets summary chart background color.

Example

Sets summary chart background color to white.

```
ThisDocument.AWXRep321.ChartBackColor=RGB(255,255,255)
```

ChartLegendHeight

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartLegendLeft

ChartLegendTop

Type: Short

Description

These properties determine the location of the corresponding chart element within the chart area. The coordinates are given in percents of the total width or height of the chart area, assuming that the coordinate origin (point {0, 0}) is located at the left top corner of the chart area.

ChartLegendVisible

Type: Boolean

Description

These properties set/get the visibility of chart legend and/or title. Setting either property to FALSE will hide the corresponding chart element.

Default value: TRUE

ChartLegendWidth

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartPlotHeight

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartPlotLeft

ChartPlotTop

Type: Short

Description

These properties determine the location of the corresponding chart element within the chart area. The coordinates are given in percents of the total width or height of the chart area, assuming that the coordinate origin (point {0, 0}) is located at the left top corner of the chart area.

ChartPlotWidth

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartTitle

Type: String

Description

Sets/gets a string value that appears in the summary chart title area.

Example

```
ThisDocument.AWXRep321.ChartTitle="Alarm Summary By Tags"
```

ChartTitleHeight

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartTitleLeft

ChartTitleTop

Type: Short

Description

These properties determine the location of the corresponding chart element within the chart area. The coordinates are given in percents of the total width or height of the chart area, assuming that the coordinate origin (point {0, 0}) is located at the left top corner of the chart area.

ChartTitleUpdateOnDataSelect

Type: Boolean

Description

Sets/gets the value of a flag that determines whether the chart title is updated when user clicks on a data series (either bar or pie segment) inside the plot. If set to TRUE (default), the name of selected data series is attached to the chart title.

ChartTitleVisible

Type: Boolean

Description

These properties set/get the visibility of chart legend and/or title. Setting either property to FALSE will hide the corresponding chart element.

Default value: TRUE

ChartTitleWidth

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartType

Type: Integer

Description

Sets/gets the type of alarm summary chart. Can be one of the following:

0 - bar type

1 - pie type

Example

Sets summary chart to pie type.

`ThisDocument.AWXRep321.ChartType=1`

Columns

Type: String

Description

Sets/gets comma-separated list of database columns to be shown on grid.

Example

This line configures control to show only EventTime, Tag and Message columns.

```
ThisDocument.AWXRep321.Columns="EventTime, Tag, Message"
```

Remarks

If database column name contains one or more space characters, the column name should be enclosed in square brackets, as in this example:

```
ThisDocument.AWXRep321.Columns="[Column 1], [Column 2]"
```

ConnectionString

Type: String

Description

Sets/gets ADO connection string used by AWXReport32 control to access data.

Example

```
Dim strConnString as String  
strConnString="Provider=Microsoft.Jet.OLEDB.4.0; _  
Data Source=c:\alarms\AWXLog32.mdb; _  
User ID=Admin; Password=SYSTEM"  
ThisDocument.AWXRep321.ConnectionString = strConnString
```

DisplayActiveFilterName

Type: Boolean

Description

Sets/gets Boolean value that controls whether the active filter(s) name(s) is(are) shown on the grid title bar.

Example

```
ThisDocument.AWXRep321.DisplayActiveFilterName = TRUE
```

DisplayGridTitle

Type: Boolean

Description

Sets/gets Boolean value that controls the visibility of the grid title bar.

Example

```
ThisDocument.AWXRep321. DisplayGridTitle = TRUE
```

EnableToolbar

Type: Boolean

Description

Setting this property to FALSE prohibits the floating toolbar to pop up when user double-clicks inside the control.

Default value - TRUE.

Example

To disable tools

```
ThisDocument.AWXRep321.EnableTools=FALSE
```

FilteringString

Type: String

Description

Sets/gets SQL WHERE statement that determines filtering of alarm data.

Example

These statements configure control to show only Humidity alarms or Pressure alarms.

```
Dim strFiltrStmt as String  
strFiltrStmt="WHERE (Tag='Humidity' OR Tag='Pressure') "  
ThisDocument.AWXRep321. FilteringString = strFiltrStmt
```

Font

Type: StdFont

Description

Returns a reference to the control's default font.

Example

To set control's default font to Arial, 14 pt., use the following statements:
With ThisDocument.AWXRep321.Font
.Name="Arial"
.Size=14
End With

ForeColor

Type: OLE_COLOR

Description

Sets/gets Forecolor property of data grid.

Example

ThisDocument.AWXRep321.ForeColor=RGB(0,0,0)

GridBackColor

Type: OLE_COLOR

Description

Sets/gets color value used to paint the gaps between rightmost column and right edge of the grid, last row and bottom edge of the grid, and row headers column (the 'back area' of the grid).

Example

ThisDocument.AWXRep321.GridBackColor =RGB(0,127,0)

GridBorderStyle

Type: Long

Description

Sets/gets the style of data grid border. Can be one of the following:

- 0 - no border
- 1 - fixed single border style

Example

`ThisDocument.AWXRep321.GridBorderStyle=1`

GridDefRowHeight

Type: Integer

Description

Sets/gets the default row height for data grid.

Example

`ThisDocument.AWXRep321. GridDefRowHeight=16`

GridHorizLinesColor and GridVertLinesColor

Type: OLE_COLOR

Description

Sets/gets color for horizontal or vertical grid lines.

Example

`ThisDocument.AWXRep321. GridHorizLinesColor
=RGB(127,127,127)`

`ThisDocument.AWXRep321. GridVertLinesColor =
RGB(127,127,127)`

GridHorizLineStyle and GridVertLineStyle

Type: Integer

Description

Sets/gets style for horizontal or vertical grid lines. Can be one of the following:

- 0 - no line
- 1 - solid line
- 2 - dot line
- 3 - dashdot line
- 4 - dashdotdot line

Example

```
ThisDocument.AWXRep321. GridHorizLinesStyle =0  
ThisDocument.AWXRep321. GridVertLinesStyle = 2
```

GridHorizLinesWidth and GridVertLinesWidth

Type: Integer

Description

Sets/gets the widths for the horizontal or vertical grid lines when corresponding grid line style (property **GridHorizLinesStyle** or **GridVertLinesStyle**) is set to 1 - solid line. For all other line styles, these properties must be set to 1.

Example

Sets both horizontal and vertical grid lines to solid line style, 2 pixels wide.
ThisDocument.AWXRep321. GridHorizLinesStyle =1
ThisDocument.AWXRep321. GridVertLinesStyle = 1
ThisDocument.AWXRep321. GridHorizLinesWidth =2
ThisDocument.AWXRep321. GridVertLinesWidth = 2

GridTitleBackColor

Type: OLE_COLOR

Description

Sets/gets the background color for the grid's title bar.

Example

Sets title bar background color to dark blue.
ThisDocument.AWXRep321. GridTitleBackColor =GRB(0,0,127)

GridTitleTextColor

Type: OLE_COLOR

Description

Sets/gets the text color for the grid's title bar.

Example

Sets grid's title bar text color to yellow.
ThisDocument.AWXRep321. GridTitleBackColor
=GRB(255,255,0)

HeaderColor

Type: *OLE_COLOR*

Description

Sets/gets the background color for grid's column headers.

Example

Sets color of grid's column header to dark green.

```
ThisDocument.AWXRep321.GridTitleBackColor =GRB(0,127,0)
```

IsConnected

Type: *Boolean*

Description

This is read-only property, indicating that control is connected to the database.

Example

```
If ThisDocument.AWXRep321.IsConnected Then  
  `` do something here ``  
End If
```

NoSorting

Type: *Boolean*

Description

Sets/gets **Boolean** value that indicates whether records sorting has been enabled or not. By default, this property is set to TRUE (sorting disabled), providing best data retrieval performance.

Example

To enable sorting:

```
ThisDocument.AWXRep321.NoSorting=FALSE
```

QueryText

Type: **String**

Description

Sets/gets the SQL statement used to populate data grid in detailed report mode.

Example

```
Dim strQry as String
strQry=ThisDocument.AWXRep321.QueryText
MsgBox strQry
```

Remarks

Though this property is read/write, it is not recommended to set **QueryText** directly. Instead, you can set the **Columns**, **FilteringString**, and **SortingString** properties, then the value of **QueryText** will be automatically updated.

QueryTextSummary

Type: String

Description

Sets/gets the SQL statement used to populate data grid in alarm summary mode.

Example

```
Dim strQrySmr as String
strQrySmr=ThisDocument.AWXRep321.QueryTextSummary
MsgBox strQrySmr
```

Remarks

Though this property is read/write, it is not recommended to set **QueryTextSummary** directly. Instead, you can set the **SummaryColumn**, **FilteringString**, **SortingString**, or **SummaryFieldSortDir** properties. The value of **QueryTextSummary** will then be automatically updated.

ReportFileAlarms

FileAlarmsCondensed

FileAlarmsEvents

ReportFileEvents

ReportFileEventsCondensed

These properties should be considered obsolete in version 6.1. They are supported only for backwards compatibility.

RowDividerStyle

Type: String

Description

Sets/gets the appearance of row divider on data grid (horizontal grid lines).
Can be one of the following:

- 0 - No dividers
- 1 - Black line
- 2 - Dark gray line
- 3 - Raised
- 4 - Inset
- 5 - Use ForeColor
- 6 - Light gray line

This property can be used to set the appearance of both vertical and horizontal grid lines in one statement. The same can be done using the **GridHorizLineColor** and **GridVertLinesColor** properties independently.

Example

To set Black row divider:
`ThisDocument.AWXRep321.RowDividerStyle=1`

ShowSummaryChart

Type: Boolean

Description

Sets/gets the **Boolean** value that controls how alarm records are reported.
Use this property to toggle between **Chart** and **Grid Display** modes.

Examples

This statement brings control into **Chart Display** mode:
`ThisDocument.AWXRep321.ShowSummaryChart=TRUE`

This statement brings control into **Grid Display** mode:
`ThisDocument.AWXRep321.ShowSummaryChart=FALSE`

ShowSummaryGrid

Type: Boolean

Description

Sets/gets the **Boolean** value that controls how alarm records are reported in **Grid Display** mode. Use this property to toggle between **Detailed** and **Summary** views in **Grid Display** mode.

Remarks

If control is in the **Chart** mode, setting **ShowSummaryGrid** property will not have immediate effect. Only after **ShowSummaryChart** property is switched to FALSE (**Grid** mode), will the desired grid view be displayed.

Examples

These statements bring the control into **Detailed Grid Display** mode:
ThisDocument.AWXRep321. ShowSummaryChart =FALSE
ThisDocument.AWXRep321. ShowSummaryGrid =FALSE

This statement brings the control into **Summary Grid Display** mode:
ThisDocument.AWXRep321. ShowSummaryGrid =TRUE

SortByDbClick

Type: Boolean

Description

Sets/gets **Boolean** value indicating that records will be sorted by values in the particular grid's column when user double-clicks on that column's header. Subsequent double clicks on the same column header will reverse sorting direction.

Remarks

The sorting action will occur only if the **NoSorting** property is set to FALSE.

Example

ThisDocument.AWXRep321. NoSorting =FALSE
ThisDocument.AWXRep321. SortByDbClick=TRUE

SortingString

Type: String

Description

Sets/gets SQL ORDER BY statement that determines sorting of alarm data.

Example

Configures control to sort alarm data by EventTime in ascending order.
Dim strSortStmt as String
strSortStmt="ORDER BY EventTime ASC "
ThisDocument.AWXRep321.SortingString = strSortStmt

SummaryColumn

Type: String

Description

Sets/gets database column name, by which the alarm records are grouped for **Summary Grid** view and **Summary Chart**. By default, this is "[Tag]" column.

Example

This statement changes the database column to summarize records by "Area":
ThisDocument.AWXRep321.SummaryColumn="[Area]"

SummaryFieldSortDir

Type: Integer

Description

Sets/gets sort direction for the **Summary** field on the **Summary Grid** view or **Summary Chart**. Can be one of the following:
0 - ascending
1 - descending

Example

This statement changes the **Summary** field sort direction to descending:
ThisDocument.AWXRep321.SummaryFieldSortDir=1

TableName

Type: String

Description

Sets/gets name of the current database table.

Example

ThisDocument.AWXRep321.TableName="EventLog"

ToolsVisible

Type: Boolean

Description

Sets/gets visibility of toolbar.

Example

To show toolbar:

```
ThisDocument.AWXRep321.ToolsVisible=TRUE
```

ToolXPos

Type: Integer

Description

Sets/gets the X coordinate of toolbar's left top corner (in device units).

Example

```
ThisDocument.AWXRep321.ToolXPos=255
```

ToolYPos

Type: Integer

Description

Sets/gets the Y coordinate of toolbar's left top corner (in device units).

Example

```
ThisDocument.AWXRep321.ToolYPos=100
```

Available Control Methods

AboutBox

Shows **About** dialog box.

Example

```
ThisDocument.AWXRep321.AboutBox
```

Connect As Boolean

Forces the control to establish connection to database, using the information set in **ConnectionString** and **TableName**. If control is already connected, the current connection will be closed and a new connection will be established. This method can be used in VBA scripts automatically to switch the report ActiveX between different databases.

Return Value

On success, returns TRUE; if for any reason connection cannot be established, returns FALSE.

Example

```
Dim strConnString as String
strConnString="Provider=Microsoft.Jet.OLEDB.4.0; _
    Data Source=c:\alarms\SomeOtherDatabase.mdb; _
    User ID=Admin; Password=SYSTEM"
ThisDocument.AWXRep321.ConnectionString = strConnString
ThisDocument.AWXRep321.TableName="SomeOtherTable"
ThisDocument.AWXRep321.Connect
    ThisDocument.AWXRep321.Refresh
```

GetColumnHeaderText(iColIndex As Integer) As String

Returns the header text for the grid column specified by iColIndex argument. The iColIndex argument is 1 - based index of the column, counting from the left of the grid. In **Summary Grid** view, iColIndex can be either 1 or 2; in **Detailed Grid** view, it can be from 1 to the maximum number of entries in the **Columns** property

Example

```
Dim strHeader1
    strHeader1=ThisDocument.AWXRep321.GetColumnHeaderT
ext(1)
```

GetColumnWidth(iColIndex As Integer) As Integer

Returns the width of the column specified by iColIndex argument. The iColIndex argument is 1 - based index of the column, counting from the left of the grid. In **Summary Grid** view, iColIndex can be either 1 or 2; in **Detailed Grid** view, it can be from 1 to the maximum number of entries in the **Columns** property. The width is given in logical units, depending on the selected font.

Example

Retrieves the current width on the leftmost column:

```
Dim cx1 As Integer  
cx1=ThisDocument.AWXRep321.GetColumnWidth(1)
```

LaunchReport(iTemplateIndex As Integer) As Integer

Creates Crystal Report based on the template specified by iTemplateIndex parameter. The value of argument iTemplateIndex can be 0 through 4 (there might be up to 5 different Crystal Report templates).

Return value

On success, returns 0; if a template for certain index does not exist or is not configured (in which case the **Crystal Report** button on the toolbar at the position corresponding to that index, counting from left, is grayed), return value is (-1).

Example

Creates Crystal Report based on the third template from the left, as seen on the **Reporter's** toolbar:

```
If ThisDocument.AWXRep321.LaunchReport(2)=-1 Then  
    MsgBox " It looks like the template  
specified is not configured"  
End If
```

Print

Description

Prints detailed alarm data, alarm summary, data or alarm summary chart depending on the current state of AWXReport32 control.

Example

```
ThisDocument.AWXRep321.Print
```

Refresh

Description

Updates alarm data by requering the database.

Example

```
ThisDocument.AWXRep321.Refresh
```

SetColumnHeaderText(iColIndex As Integer, strNewText As String)

Sets the header text for the grid column specified by iColIndex argument. The iColIndex argument is 1 - based index of the column, counting from the left of the grid. In **Summary Grid** view, iColIndex can be either 1 or 2; in **Detailed Grid** view, it can be from 1 to the maximum number of entries in the **Columns** property.

Example

```
ThisDocument.AWXRep321.SetColumnHeaderText(1, "Column1")
```

SetColumnWidth(iColIndex As Integer, iNewWidth As Integer)

Sets the width of the column specified by iColIndex argument to iNewWidth. The iColIndex argument is 1 - based index of the column, counting from the left of the grid. In **Summary Grid** view, iColIndex can be either 1 or 2; in **Detailed Grid** view, it can be from 1 to the maximum number of entries in the **Columns** property. The width is given in logical units, depending on the selected font.

Example

Doubles the current width on the leftmost column:

```
Dim cx1 As Integer  
cx1=ThisDocument.AWXRep321.GetColumnWidth(1)  
ThisDocument.AWXRep321.SetColumnWidth(1, 2*cx1)
```